Autoren: Matthieu und ag_bg von www.synology-forum.de

Synology DiskStations – Kleiner Guide

Ein kleiner Guide durch die Tipps und Tricks rund um die DiskStations der Firma Synology.

13.03.2009

Synology DiskStations – Kleiner Guide

Inhalt

Vor dem Lesen 5
Das Autorenteam 5
Quellen 5
Danksagung 5
Vorwort von Matthieu 5
Haftung 6
Hinweise zur Verwendung 6
Warenzeichen 6
1. Grundlegende Standards und ihre Verwendung auf einer DiskStation 7
a. http/https 7
b. FTP (File-Transfer-Protokoll) 7
c. Telnet/SSH 8
d. PHP 9
e. (My-)SQL 9
f. Ports 9
g. DynDNS 10
h. Raids 10
i. Jumboframes 11
j. RSS 11
k. POP3, SMTP, IMAP 12
2. Welche Version ist für mich die richtige? (Vor dem Kauf) 14
3. Der erste Betrieb 15
4. Die Bestandteile der Firmware und kleine „Tricks“ 16
a. Webstation 16
i. Zugriffschutz eigener Seiten mit .htaccess 16
ii. Suchmaschinen abweisen 17
iii. Alle Zugriffe auf den Webserver loggen 17
iv. Eigener “404-Error” 18
b. PhotoStation 18
c. FileStation 19
l. Zugriff über FTP 19

d. DownloadStation 19
e. AudioStation 19
f. iTunes-Server 20
g. SurveillanceStation 23
h. MailStation 23
i. MailStation als vollwertiger Mail-Server 24
i. Grundlegendes zum Thema Sicherheit 25
5. Backup 27
a. Externe Festplatten-Sicherung 27
b. Netzwerksicherung 27
c. PC-Sicherung 27
d. Einstellungen etc. 28
6. Das Linux auf der DiskStation 29
a. Warum Linux? 29
i. Geschichte von Linux 29
ii. Linux unter GPL-Lizenz 30
iii. Linux ist frei verfügbar 30
iv. Linux ist modular 30
vi. Linux ist sicher 30
b. Die Verzeichnisse des Synology-Linux 31
c. IPKG 32
i. Die Installation 32
d. Zugriff über SSH 33
e. Der/Die Apache-Webserver 34
7. CMS4DS 35
a. Wozu ein neues CMS? 35
b. Das Konzept 35
c. Der Aufbau 36
d. Die html-Seiten im Detail 37
i. Header.html 37
ii. footer.html 38
iii. bottom.html 38
e. Stylesheet 39
f. Javascript 39
f. CMS4DS+SQL=? 43

g. RSS 46
h. Tools 47
8. Andere Anwendungen auf der DS („3rd-Party-Apps“) 49
a. Vor dem Modden! 49
b. Toolchain 49
c. Bekannte 3rd-Party-Anwendungen 49
9. Nützliche Links 50

Vor dem Lesen

Das Autorenteam

Derzeit arbeiten wir zu zweit an diesem Guide:

- Matthieu
- Ag_bg (iTunes-Server)

Wer über gute Kenntnisse zu den hier besprochenen Themen verfügt und ein wenig Zeit übrig hat sowie gerne schreibt, kann sich bei uns im Synology-Forum melden.

Quellen

- http://synology-wiki.de
- http://synology-forum.de
- http://de.selfhtml.org
- http://de.wikipedia.org
- Synology Benutzerhandbuch
- *Krzysztof Janowicz*: Sicherheit im Internet. 3. Auflage, O'Reilly Verlag, Köln, 2007, ISBN: 978-3-89721-715-7
- *Jon Masters, Richard Blum*: Professional Linux Programming. Wiley Publishing, Indianapolis, 2007, Reprint by Wiley India, ISBN: 10-81-265-1204-0
- Beiträge im Forum *(http://synology-forum.de)* von verschiedenen Nutzern, insbesondere itari und Pax90
- http://itari.syno-ds.de und http://itari.synology-ds.de

Danksagung

Ich möchte mich bei den Nutzern des deutschen Synology Forums (http://synology-forum.de) für ihre Mithilfe und Fehlerkorrektur sowie ihre Ideen bedanken.

Außerdem möchte ich mich bei den Nutzern des Forums bedanken, welche vor der offiziellen Veröffentlichung über den Text geschaut haben und mir mit Rat und Tat zur Seite standen:

- MJFox
- Manuel_bo

Vorwort von Matthieu

Als ich die ersten Seiten geschrieben hatte und dann mein Vorhaben im Forum ankündigte, waren viele zunächst sehr skeptisch. Einige sagten, ich sollte lieber ins Wiki, unserer Wissensdatenbank, schreiben anstatt meine Zeit zu vergeuden. Einige meiner Texte finden sich nun auch in besagtem Wiki, doch ich bin nicht von meinem Vorhaben abgerückt, eine eigene Datei zu erstellen, welche grundsätzlich zum Offline-Lesen und drucken gedacht hat. Im Gegensatz zum Wiki enthält dieses „Mini-Buch" aber nicht nur Informationen und Modifikationen für die DiskStation, sondern auch über deren Umfeld: das Internet. Denn wer verstehen möchte, womit er es eigentlich zu tun hat, dem darf das passende Hintergrundwissen nicht fehlen. Dazu zählt zum einen das, woraus das Internet gemacht ist: Protokolle, Programmiersprachen, … Zum anderen beschäftige ich mich auch mit dem Linux, welches den Synology-Produkten als Betriebssystem („Firmware") dient und der Entwicklung eigener Anwendungen. Denn warum nicht selbst Hand anlegen, wenn es so einfach ist?

Dieser Text eignet sich für Einsteiger und Fortgeschrittene gleichermaßen und begleitet den Leser von Kauf und Inbetriebnahme, bis hin zu den kleinen Kniffen, die das letzte aus der DiskStation herausholen.

Haftung

Dieses Dokument ist keine Publikation einer offiziellen Stelle. Das Dokument erhebt weder den Anspruch auf Vollständigkeit noch auf Fehlerfreiheit. Für Fehler in diesem Dokument übernimmt der Autor keinerlei Haftung. Die Verwendung der Modifikationen erfolgt ausdrücklich auf eigene Gefahr.

Hinweise zur Verwendung

- Dieser Guide wurde zur Firmware „DSM2.1-0832“ geschrieben. Sollten sie eine andere Firmware verwenden kann es gewisse Unterschiede geben.
- Sämtliche Links welche sich auf die DiskStation beziehen (z.B. http://DiskStation:5000/) funktionieren in ihrem Netzwerk wahrscheinlich nicht. Die Bezeichnung „DiskStation“ müssen sie daher entweder durch die IP oder den Namen ihrer DiskStation ersetzen. Wenn sie im LAN arbeiten empfiehlt es sich nicht, die DynDNS-Adresse zu verwenden, da einige Router dies nicht korrekt umsetzen können.

Warenzeichen

Synology und andere Namen von Synology-Produkten sind geschützte oder eingetragene Warenzeichen der Synology Inc. Microsoft, Windows, Windows 2000 und Windows XP sind Warenzeichen der Microsoft Corporation. Mac, Mac OS und Mac OS X sind Warenzeichen von Apple Computer, Inc., eingetragen in den USA und in anderen Ländern. Andere hier erwähnte Produkte und Firmennamen sind Warenzeichen ihrer jeweiligen Inhaber.

1. Grundlegende Standards und ihre Verwendung auf einer DiskStation

a. http/https

Zu Beginn möchte ich auf ein paar grundlegende Protokolle eingehen, ohne welche das Internet wie wir es kennen, nicht existieren würde. Ein Protokoll ist ein Satz von Regeln, wie Daten übertragen werden sollen. Ohne einen solchen Standard können zwei Geräte zwar verbunden sein, aber nicht miteinander kommunizieren. Das grundlegendste Protokoll ist TCP/IP. Computer welche dieses Protokoll nicht unterstützen, sind nicht in der Lage, in einem Netzwerk zu arbeiten. Diese beiden Protokolle dienen dazu, eine Verbindung zwischen zwei Geräten (z.B. zwei Computern in einem Netzwerk, oder einem Computer und einem Server im Internet) aufzubauen.

Nachdem eine Verbindung zwischen den beiden Geräten aufgebaut wurde, tritt ein anderes Protokoll in Kraft um Daten zu übertragen. Im Falle von Webseiten ist dies meistens http. Das *hyper-Text-Transfer-Protokoll* bietet eine große Vielfalt der möglichen zu übertragenden Medien. So kann es einfachen Text genauso bereitstellen, wie Farben, Bilder oder auch ganze Dateien. In den DiskStations kommt http beim Webserver, welcher über http://DiskStation/ erreicht werden kann, und bei der Administration über das Interface, welches per http://DiskStation:5000 erreicht werden kann, zum Einsatz. Hinter dem Doppelpunkt werden Ports angegeben. Zu diesen möchte ich aber später mehr sagen. Der größte Nachteil des http-Protokolls, die fehlende Verschlüsselung und somit geringe Sicherheit. Dies bringt uns zu https. Hier wird die gesamte Übertragung verschlüsselt und sollte damit ausschließlich für wichtige Informationen, wie Passwörter oder persönliche Informationen genutzt werden. Synology®-Produkte lassen sich ansprechen, indem wir einfach das „http“ am Anfang gegen „https“ eintauschen (https://DiskStation/). Bei der Administrationsoberfläche hat sich hingegen auch der Port geändert: https://DiskStation:5001/. Aufgrund der Verschlüsselung dauert die Übertragung natürlich etwas länger, doch dank der Geschwindigkeit heutiger Internetverbindungen und aktueller Computer ist dieser Unterschied fasst nicht mehr auszumachen. Betreiben sie auf ihrer DiskStation Webseiten, welche sensible Informationen, wie Passwörter oder persönliches nutzen, sollten daher ausschließlich per https angesprochen werden.

b. FTP (File-Transfer-Protokoll)

Im Gegensatz zu http, kann ftp keine Internetseiten übertragen. Seine Stärken liegen in der Übertragung von Dateien. Während http ein „Allround-Talent“ ist und daher keine besonders guten Geschwindigkeiten bei Downloads erreichen kann, holt ftp deutlich mehr aus dem möglichen heraus. Die meisten gängigen Browser unterstützen das ftp-Protokoll und liefern über ftp://DiskStation/ meist einen Überblick über ihre Dateien. Doch dies ist generell sehr Text-orientiert und unübersichtlich. Daher sind separate Programme, wie das frei erhältliche WinSCP deutlich besser für diese Zwecke geeignet. Diese Programme bieten eine Windows-Explorer-ähnliche Oberfläche und machen den Dateitransfer per simplen Drag-und-Drop möglich.

Doch ähnlich wie http, hat auch ftp mit der fehlenden Sicherheit zu kämpfen. Aufgrund der fehlenden Verschlüsselung braucht man nur den Internetverkehr mitzuschneiden und kann dann recht einfach die Anmeldedaten auslesen und missbrauchen. Die Lösung heißt sftp. Dieses Protokoll nutzt als Grundlage SSH (womit wir uns als nächstes beschäftigen) und verwendet dessen Verschlüsselung um auf Dateien zuzugreifen. Aufgrund der besseren Ausnutzung der Verbindung sollten sie immer als Download-Link für ihre Benutzer ftp statt http nutzen. Ein Beispiel für die längeren Zeiten ist die Nutzung der FileStation, welche auf http statt (s)ftp setzt. Bleibt nur noch das Problem, dass die DiskStation kein sftp unterstützt. Stattdessen kann man verschlüsselt per SSL/TLS zugreifen.

c. Telnet/SSH

Telnet stammt aus der Zeit von Unix und wird verwendet um auf andere Rechner per Remote (engl. Fernbedienung) zuzugreifen. Der Nutzer arbeitet dann in einer sogenannten Shell, welches meist ein Textfenster mit meist weißer Schrift auf schwarzem Hintergrund darstellt. Wer derartige Technologien für veraltet hält, dem sollte gesagt sein, dass Microsoft in Windows 7 sich, nachdem man der hauseigenen Shell schon den Rücken gekehrt hatte, wieder zum standardmäßigen Einbau entschlossen hat, damit Netzwerkadministratoren direkt auf andere Windows-Rechner im Netzwerk zugreifen können, ohne durch die ganze Firma zu rennen. Diese Technologie beherrscht natürlich auch der UNIX-Abkömmling Linux, welcher ja bekanntlich auf den Synology®-Produkten eingesetzt wird. Auch für Telnet gibt es natürlich wieder einen Client, der einem die Arbeit erleichtert. Sehr bekannt ist Putty, mit welchem man recht einfach eine Verbindung aufbauen kann. Dazu geben sie die Adresse des zu kontrollierenden Gerätes ein, wählen die Verbindungsart (Telnet, SSH, …)(belassen den Port bei 22 bzw. 23) und bestätigen. Nach ein paar Sekunden sollten sie dann das beschriebene, simple Fenster sehen. Zuerst werden sie nach dem Benutzernamen gefragt. Aus Erfahrung würden sie hier natürlich „admin“ eintragen. Doch wer schon mal mit Linux gearbeitet hat, weiß, dass der Nutzer mit den meisten Rechten hier „root“ heißt. Das Passwort unterscheidet sich allerdings nicht von dem des „admin“. Was genau sie nun machen können, werden wir später genauer erörtern. Wer trotzdem schon mal sehen will, was man machen kann, tippt „help“ in die Konsole. Seien sie allerdings vorsichtig, denn sie arbeiten nun im absoluten Kern des Betriebssystems. Ein falscher Befehl und ihr System ist hinüber. Aus genau diesem Grund ist es sehr gefährlich, wenn Eindringlinge auf diesem Weg Zugriff auf ihr System bekommen. Insbesondere Telnet hat mit besonders hohen Sicherheitsproblemen zu kämpfen, da es keine Verschlüsselung einsetzt und die gesamte Übertragung im Klartext erfolgt. Aus diesem Grund wurde SSH entwickelt. Diese *Secure Shell* verschlüsselt effektiv den gesamten Verkehr. Daher sollte man NIE über Telnet aus dem Internet auf ihre DiskStation zugreifen. Da wir, wie bereits gesagt, im Kern des Linux werkeln, ist es auch ein umso größeres Problem, wenn ein Hacker erfolgreich eindringen kann. Daher mein Rat: Lassen sie die Option für Telnet immer aus und nutzen sie stattdessen SSH. Dieses mal geht es nicht nur um ihre Daten, sondern um ihre gesamte DiskStation Natürlich muss dabei auch wieder ein anderer Port verwendet werden. Bei uns ist das die 23.

d. PHP

Nun entfernen wir uns von den Protokollen. Der Unterschied besteht nunmehr in der Programmiersprache, die zur Programmierung verwendet wird. Dort wo html aufhört, macht PHP weiter. Es bietet sehr flexible Möglichkeiten zum erstellen dynamischer Internetauftritte. Auf der anderen Seite kann PHP kein html ersetzen. Im Gegenteil: PHP kann nicht von Browsern interpretiert werden, da viele Informationen welche PHP nutzt, nur für den Server (in unserem Fall unsere DiskStation) zugänglich sind. Bevor die Seite daher übertragen wird, wandelt der Server den PHP-Code in normale html-Seiten um. Trotzdem kann PHP ganz einfach mit html in einer Datei eingesetzt werden, ohne das eine neue Datei benötigt wird. Die Macher von PHP vereinen in ihrem Konzept die Vorteile von JavaScript (einfach zu erlernen, in selber Datei verwendbar) und Perl (vielfältige Möglichkeiten, teilweise aber sehr kompliziert). Auf unserer DiskStation bietet PHP neben vollen Linux-Programmen die einzige Möglichkeit mit Datenbanken zu arbeiten. Es wird außerdem frei entwickelt und besitzt keine Lizenzkosten. Außerdem ist für Einsteiger schnell zu erlernen und trotzdem sehr leistungsfähig. Der Webserver der DiskStation unterstützt PHP nach Aktivierung ohne Probleme.

e. (My-)SQL

Recht häufig müssen Internetseiten auch größere Mengen von Daten erstellen und verwalten. Als Beispiel dienen dazu Online-Shops. Hierzu eignen sich am einfachsten Datenbanken. Das wohl bekannteste System ist SQL. SQL an sich ist allerdings kein Programm sondern nur ein Standard. Microsoft und andere Hersteller vertreiben verschiedene SQL-Server. Der beliebteste ist allerdings das kostenlose MySQL. Es bietet nicht zuletzt auch vergleichsweise hohen Komfort bei geringer Ressourcen-Nutzung. MySQL an sich kann ganz einfach aktiviert werden. Jedoch fehlt der DiskStation jegliche Möglichkeit der Administration dieses Dienstes. Daher wird recht verbreitet PHPMyAdmin eingesetzt. Dieses kann extra installiert werden (siehe www.synology-wiki.de).

f. Ports

Um dies ein wenig einfacher zu erklären, stellen sie sich ihre DiskStation als Haus vor. Ihr Router, welcher den Internetzugang regelt, bringt fast immer auch eine eigene Firewall mit. Die DiskStation selber besitzt keine. Die Firewall stellen wir uns jetzt wie einen Zaun um das Haus vor (wer will kann sich auch einen brennenden Zaun vorstellen ;-)). Um verschiedene Internetdienste, wie Webserver, E-Mail-Server, Telnet, ftp, etc. alle über dieselbe Adresse erreichen zu können ohne für jeden eine neue zu belegen, wurden Ports erschaffen. Diese Nummer, welche hinter der IP mit einem Doppelpunkt abgetrennt wird, weißt auf den Dienst hin. Jeder Port soll daher durch ein Tor in unserem Zaun veranschaulicht werden. Normalerweise sind alle Ports geschlossen, sodass keine Sicherheitslücken entstehen. Wer nun allerdings einen Server betreibt, muss bestimmte Tore öffnen um den Verkehr ungehindert fließen zu lassen. Andernfalls würden die Dienste, welche hinter den Ports warten, nicht im Internet zur Verfügung stellen. Die bekanntesten Ports sind z.B. 80 für http-

Server bzw. 443 für das sichere https sowie 20/21 für ftp. Wer einen Dienst nicht im Internet benötigt, sondern nur im eigenen Netzwerk, sollte den Port nicht öffnen, da ja jeder ein kleines Sicherheitsrisiko darstellt. Ein offenes Tor wird halt gerne von unerwünschten Leuten genutzt, wenn sich dahinter etwas Begehrenswertes befindet. Selbst wenn das Begehren nur die reine Zerstörung ist. Welcher Port von welchem Dienst der Synology® DiskStations verwendet wird, kann man hier nachlesen: http://www.synology.com/enu/support/help-page.php?q_id=299

g. DynDNS

Wer schon einmal versucht hat, seine DiskStation von außen zu erreichen, wird mit einem Problem konfrontiert: Moderne DSL-Zugänge wechseln ihre IP-Adressen laufend. Selbst wer nun sagt, dass er seinen Router nie vom Netz trennt und daher ja keine neue IP bekommen kann, irrt sich. Mindestens einmal pro Tag wird in diesem Fall der Router „Zwangs-Neu-Verbunden“. Um einen Weg aus dieser Kriese zu finden, gibt es DynDNS. DNS steht für „Domain-Name-System“. Dieses System regelt, welche IP zu welcher Adresse gehört. Das Kürzel „Dyn“ steht wiederum für „dynamisch“. Heraus kommt eine dynamische Namens-Adresse. Anbieter für dieses System gibt es einige. Bei den kostenlosen Anbietern wie dem amerikanischen „dyndns.org“ muss man allerdings mit der Einschränkung leben, nur eine Sub-Domain zu besitzen (z.B. http://me.dyndns.org). Wer dagegen bereit ist, ein wenig Geld auszugeben, kann preiswert sogar eine echte .de-Domain bekommen. (z.B. selfhost.de).

h. Raids

Bevor wir unsere DiskStation jetzt in Betrieb nehmen können, müssen wir uns mit einem weiteren Thema beschäftigen: Was passiert, wenn die Festplatte in die Knie geht? Alle Daten verloren? Da solche Ausfälle in der IT-Branche unbezahlbar sind, gibt es natürlich ein Gegenmittel. Sogenannte Raids kopieren alle Dateien auf verschiedene Festplatten, um im Falle eines Defektes ohne Datenverlust weiter arbeiten zu können. Bei den größeren DiskStations, welche Hot-Swap-fähig sind, können wir sogar im laufenden Betrieb die defekte Platte austauschen. Komplette Sicherheit kann allerdings nicht gewährleistet werden, denn wenn man eine Datei löscht, fehlt sie danach nicht nur auf einer Platte, sondern auf allen. Doch mit Backups, welche in diesem Fall helfen können, werden wir uns noch früh genug auseinander setzen. Da es nicht einfach ist, die Raids im Nachhinein zu ändern, sollte man von Anfang an die Konfiguration wählen, welche für einen den meisten Sinn ergibt.

Hier ein kurzer Überblick, über die von den DiskStations unterstützten Möglichkeiten:

Raid-Level	Benötigte Festplatten	Tolerierte, gleichzeitige Ausfälle	Kapazität des neuen, logischen Laufwerks (Laufwerke/Größe)
Kein Raid	1	0	1/1
Raid-0	2,3,4,5	0	5/5
Raid-1	2	1	2/1
Raid-5	3,4,5	1	5/4

Raid-6	3,4,5	2	5/3

Kleine Erklärung: „Benötigte Festplatten“ gibt an, wie viele Festplatten verwendet werden können. „Tolerierte, gleichzeitige Ausfälle“ gibt an, wie viele Festplatten Schaden nehmen dürfen, ohne das man Angst vor Datenverlust haben muss. „Kapazität“ gibt an, wie groß das neue, virtuelle Laufwerk wird (z.B 5/3 Wenn sie fünf Festplatten einsetzen, hat das neue Laufwerk die Größe von drei Laufwerken).

Beachten sollte man, dass die Platte mit der geringsten Kapazität immer die Kapazität aller anderen vorgibt. (z.B.: Sie haben 4 Festplatten a 300GB und eine mit 250 GB, dann ist die Kapazität am Ende so groß, als würden sie 5x250GB einsetzen.) Es empfiehlt sich also, baugleiche Festplatten zu verwenden.

i. Jumboframes

Aus den Urzeiten der Netzwerke stammt die Festlegung, dass ein Paket von Daten maximal 1.518 Bytes umfassen darf. Damals entschied man sich für eine derartige Festlegung, da Fehler in der Übertragung recht häufig vorkamen und dann das Paket erneut gesendet werden musste. Umso kleiner das Paket damals also war, umso schneller ließ es sich erneut senden. Nun hat in den letzten Jahren die Verbesserung der Netzwerktechnologie aber dazu geführt, dass Fehler viel seltener auftreten. Da die Pakete relativ klein sind und sie nun deutlich schneller eintreffen, spielt die Rechengeschwindigkeit und andere Dinge eine große Rolle. So dauert es relativ lange, bis überprüft werden konnte, ob das Paket so angekommen ist wie es sollte. Außerdem kann man sich, denke ich, gut vorstellen, warum es länger dauert eine Menge kleine Pakete zu schicken. Stellen es sie sich so vor, als würden sie umziehen. Ob sie ihren Hausrat in kleine oder große Pakete packen, macht für den LKW recht wenig unterschied. Sie können aber immer nur eine begrenzte Menge von Paketen sicher das Treppenhaus herunter bringen. Auch das verladen könnte eventuell etwas länger dauern. Somit ist der Vorteil von größeren Paketen wohl recht gut ersichtlich.

Nun aber das Problem. Die bisherigen Standards berücksichtigen Jumboframes nicht. Viele Gigabit-Switches, -Router und –Karten unterstützen zwar Jumboframes, doch das ist bei weiten nicht bei allen der Fall. Normale Jumboframes sind zwischen 1500 und 9000 Bytes groß. Doch nicht alle werden von jedem Gerät unterstützt. Alles in allem können Jumboframes das letzte Stück Performance aus ihrem Netzwerk holen, auf der anderen Seite sind sie aber auch kein Spielzeug, da Fehleinstellungen schnell das gesamte Netzwerk lahm legen können.

j. RSS

Obwohl es eigentlich, technisch gesehen, nur eine XML-Seite ist, wird der RSS-Standard immer wichtiger und immer mehr Menschen nutzen ihn um über die neuesten Ereignisse im Bilde zu sein.

Um seine Daten abzugleichen, funktioniert ein abonnierter RSS-Feed fast wie ein E-Mail-Programm. In regelmäßigen Abständen gleicht es seine Daten mit dem des Servers ab und lädt bei Änderungen das neue Material herunter.

Obwohl es nie richtig offiziell geworden ist, verwenden die meisten Seiten und Programme dasselbe Symbol um RSS zu symbolisieren.

In unserer DiskStation kommt RSS standardmäßig im Blog zum Einsatz.

Die Geschichte von RSS ist nicht ganz einfach. Da der Standard nie von jemandem lizenziert wurde oder jemand offiziell dessen Entwicklung übernommen hat, existieren verschiedene Versionen. Zunächst gab das Netscape-Netzwerk, welches der Nachrichtenteil des gleichnamigen Browsers ist, im Jahre 1999 einen Standard heraus, welcher auf RDF, eine Alternative zu XML, setzte. Diese „Rich Site Summary“ wurde aber bald durch „RDF-Site-Summary“ im Jahr 2000 abgelöst, welches inoffiziell von UserLand Software weiterentwickelt worden war und Version 0.91 darstellte. Bald veröffentlichte man auch 0.92 sowie Entwürfe für 0.93 und 0.94. Parallel zu 0.91 wurde von einer unabhängigen Entwicklergruppe die Version 1.0 geschaffen, welche wiederum auf RDF basierte und ebenfalls im Jahr 2000 veröffentlicht wurde. UserLand stellte seine Arbeiten aber nicht ein und entwickelte bis 2002 die Version 2.0 welche aber unter Kritik steht, da es nicht vollständig abwärtskompatibel zu 0.9x ist. Doch da es die aktuellste Version darstellt, setzt es sich trotzdem immer weiter durch.

Als Alternative zu RSS-Feeds haben sich Atom-Feeds etabliert, welche ihrerseits auf RSS 2.0 basieren.

k. POP3, SMTP, IMAP

Diese drei Protokolle werden von uns jeden Tag genutzt. Aber im Gegensatz zu http, fällt uns das nicht jeden Tag ins Auge. Denn einmal konfiguriert, läuft alles automatisch. Die Rede ist von E-Mail-Protokollen. In der DiskStation werden sie von der MailStation verwendet, welche als spk-Paket bei Synology vorliegt und einfach mit einem Klick installiert werden kann. Die bekanntesten sind POP3, IMAP und SMTP und werden auch von der MailStation in vollem Umfang unterstützt.

Zunächst zum PostOfficeProtocol in Version 3.0 oder kurz POP3. Dieser Vertreter beschränkt sich rein auf das Abholen von Mails. Genauer genommen unterstützt es das Auflisten, Abholen und Löschen der Mails, bietet aber keine Möglichkeit des Sendens und muss daher immer im Verbund mit einem anderen Protokoll genutzt werden. Standardmäßig verwendet POP3 den Port 110, wobei auch die MailStation keine Ausnahme ist. Wer also auch von außerhalb dieses Protokoll nutzen möchte, wird um eine Portfreigabe nicht herumkommen. Die Version eins wurde bereits 1984 veröffentlicht und es dauerte nur ein Jahr, bis die zweite Version folgte. 1988 fand dann die endgültige Version 3 seine Endfassung. Die meisten gängigen Mail-Clients unterstützen POP3 ohne Probleme.

Die Übertragung von Daten erfolgt allerdings, ähnlich wie bei FTP, komplett unverschlüsselt und im Klartext, wodurch es sehr anfällig für Sicherheitsrisiken ist. Dafür wurde POP3S geschaffen, welches über Port 995 kommuniziert und SSL/TLS für die Verschlüsselung nutzt, also ebenfalls vergleichbar mit dem von der DiskStation verwendeten FTP over SSL/TLS. Auch gibt es verschiedene Ansätze für serverseitige Sicherung, doch dort konnte sich keine Software bisher durchsetzen.

Als nächstes zu IMAP oder InternetMessageAccessProtocol. Als Portnummer wurde hier 143 registriert. Auch IMAP kennt verschiedene ältere Versionen, welche aber nicht mehr genutzt werden, weshalb eine Unterscheidung nicht nötig ist. Die Entwicklung begann 1986 mit dem Ziel, alle Nachrichten so bereitzustellen, als ob die Mails direkt auf dem Rechner liegen würden. Im Gegensatz zu POP3, bleiben die Mails komplett auf dem Server und werden nicht heruntergeladen. Gemeinsam hat es allerdings die Eigenschaft, keine Mails versenden zu können. Das die Mails auf dem Server bleiben, hat aber noch andere Nebenwirkungen. So ist der Netzwerkverkehr recht hoch und da beispielsweise die Suche vom Server erledigt werden muss, bedeutet auch für unsere DiskStation mehr Arbeit.

Wie bereits POP3, verwendet auch IMAP standardmäßig keine Verschlüsselung. Stattdessen muss die Verbindung über Port 993 gelenkt werden um eine Verschlüsselung über SSL zu gewährleisten. Man sollte die Verwendung von IMAP auf der eigenen DiskStation allerdings gründlich abwägen, aufgrund des bereits angesprochenen höheren Netzwerkverkehrs und der gesteigerten Leistungsanforderungen.

SMTP ist auf der DiskStation die einzige Möglichkeit, Mails zu versenden. Leider ist auch dem SimpleMailTrafficProtocol seine mehr als 20-jährige Geschichte anzumerken. Es verwendet Port 25 und bietet leider keine Möglichkeit der Verschlüsselung. Unglücklicherweise bietet die MailStation sogar die Möglichkeit, das größte Problem an SMTP unkorrigiert zu lassen. Denn standardmäßig sieht SMTP nicht einmal vor, dass sich die Nutzer mit Benutzername und Passwort authentifizieren müssen. Diese Funktion wurde später hinzugefügt. **Man sollte daher immer den entsprechenden Haken in der MailStation setzen!**

Wer sich allerdings gar nicht erst mit Portweitergaben und Protokollfragen herumschlagen möchte, lässt die Ports einfach geschlossen und verwendet stattdessen den RoundCube-Webmail-Client, welcher unter http://DiskStation/mail erreichbar ist. Besser ist natürlich die Verwendung von https. Für RoundCube muss nur IMAP aktiviert sein.

2. Welche Version ist für mich die richtige? (Vor dem Kauf)

Sehr häufig lese ich, dass es viele Fragen gibt, welche DiskStation nun genau die richtige für die eigene Verwendung ist. Insbesondere sind viele unschlüssig, ob sie das „+“ wirklich brauchen. Die erste Ziffer gibt jeweils die Anzahl der Festplatten an. Die letzte Ziffer steht für die Version. Zwischen den Versionen ändert sich meist nur die Hardware, welche stets schneller und leistungsfähiger wird. Die einfacheren DiskStations sind schnell überfordert. Wer daher mehr als nur die eingebauten Funktionen nutzen möchte, sollte stets zu einer „+“-Variante greifen. Mit der Einführung der 803-Firmware, zeigt sich auch bei den eingebauten Features, dass nicht viel Leistung von den kleineren Modellen erwartet werden kann, denn die MailStation wird nicht für alle Versionen verfügbar sein. Entsprechend veränderte Pakete gibt es zwar zum Download, allerdings ist dann anzunehmen, dass die Geschwindigkeit stark abnimmt. Wenn man bedenkt, dass Arbeitsspeicher und Prozessor praktisch nicht ausgebaut werden können, lohnt es sich ein wenig mehr Geld auszugeben, als dann an der geringeren Rechenleistung zu scheitern. Wer sich von der Auslastung des Arbeitsspeichers und des Prozessors einmal ansehen möchte, kann einmal die Befehle „`cat /proc/meminfo`“ und „`cat /proc/cpuinfo`“ per SSH ausprobieren (nachdem man sich für eine DiskStation entschieden hat). Eine der besten Beispiele für hohe Hardware-Anforderungen ist das bekannt Blog-System „Wordpress“, welches selbst auf einer „DS207+“ nicht immer läuft. Glücklicherweise arbeitet aber auch Synology kontinuierlich an der Firmware, sodass insbesondere die Speichernutzung ständig verbessert werden kann, wie aktuelle Updates zeigen. Die stete Verbesserung der Synology-Produkte ist eine der Hauptgründe, warum die Fangemeinde um diese Produkte so groß ist, und stetig steigt.

3. Der erste Betrieb

Bevor ich mit diesem Kapitel beginne, möchte ich darauf hinweisen, dass derartige Prozeduren von Firmware zu Firmware verschieden sein können.

Bevor sie überhaupt nur ans Stromnetz gehen können, müssen sie sich zunächst die fehlenden Festplatten besorgen. Da es häufig Probleme mit inkompatibler Hardware gibt, sollte man immer einen Blick auf die von Synology® veröffentlichten Listen werfen um einen reibungslosen Betrieb zu ermöglichen (http://synology.com/enu/support/help-page.php?q_id=130).

Der genaue Einbau der Festplatten ist von Modell zu Modell verschieden. Daher möchte ich hier wieder auf eine Internetseite verweisen und nur eine allgemeine Anleitung in Textform geben. Detaillierte Anleitungen gibt es hier: http://synology.com/enu/support/download.php

Zunächst sollte man die gekauften Festplatten vorbereiten (alles auspacken, Kabel anstecken, eventuell auch schon einen ersten Funktionstest durchführen). Anschließend muss das Gehäuse geöffnet werden und die Festplatte mit dem kürzeren Kabel zuerst befestigt werden. Sind alle Festplatten verschraubt, werden die Kabel in Position gebracht und das Gehäuse wieder zusammengesetzt und verschraubt.

Nun kann man sein neues Schmuckstück erstmals ans Stromnetz lassen, nachdem man das LAN-Kabel befestigt hat. Wenn der Start laut den LEDs Planmäßig gelaufen ist, beginnt die Arbeit am eigenen Computer. Zunächst muss die mitgelieferte CD gestartet werden. Das dort mitgelieferte Dienstprogramm kann die DiskStation recht unkompliziert mit „einem Klick“ einrichten.

4. Die Bestandteile der Firmware und kleine „Tricks“

Achtung: Bei einigen der hier beschriebenen Vorgehensweisen handelt es sich um Änderungen am System, welche von Synology nicht unterstützt werden. Alles erfolgt daher ausdrücklich auf eigene Gefahr!

a. Webstation

Die DiskStation bringt einen eigenen Webserver mit, welcher nach Aktivierung die eigene Homepage im Internet präsentiert. Doch alles der Reihe nach. Zunächst sollten sie sich vergewissern, dass die Webstation in der Administrationsoberfläche aktiviert ist und die entsprechenden Ports in ihrem Router weitergeleitet wurden (http: 80). Wer das verschlüsselte https nutzen möchte, muss dies vorher noch aktivieren. Nun erstellt die Firmware den Ordner „web“. Ihre erstellten Seiten sind ab sofort über die Adresse der DiskStation erreichbar (http://DiskStation/). Beachten müssen sie nur, dass der Webserver nach einer Datei namens „index.html“ (oder „index.php) sucht und diese zuerst öffnet. Existiert eine solche Seite nicht, bekommen sie eine Fehlermeldung. Bevor sie sich jetzt aber auf die Vielzahl der verfügbaren Programmiersprachen stürzen, müssen wir auch auf diese noch einmal kurz eingehen.

Bei den Web-Programmiersprachen wird grundsätzlich zwischen zwei Techniken unterschieden. Die erste Kategorie sind die clientseitigen Programmiersprachen. Wenn ein Nutzer auf eine Seite zugreift, welche clientseitige Programmiersprachen nutzt, wird der Klartext übertragen und dann im Browser des Nutzers in etwas Visuelles umgesetzt. Die wohl am häufigsten genutzten, clientseitigen Sprachen sind Javascript (DOM), ActiveX, Flash und Java. Diese Sprachen können sie ungehindert einsetzen, da sie keine Anforderungen an den Webserver stellen. Wenn wir uns aber die serverseitigen Sprachen ansehen, müssen wir etwas vorsichtiger sein. Denn hier muss der Webserver, in unserem Fall ein Apache (eventuell einigen vom XAMPP-Paket bekannt), den Code umsetzen und dann in html-Form an den Client schicken. Die DiskStation unterstützt ausschließlich PHP und (nach Installation) auch eine entsprechende Anbindung an MySQL. Dies sollte für die meisten Heimprojekte eigentlich genug sein. Doch der Vollständigkeit halber hier noch ein paar, nicht unterstützte Formate: JSP, ASP und Perl.

Um nun den Zugriff auf unsere neue Webpräsenz zu erleichtern, sollte man sich darüber hinaus einmal DynDNS ansehen.

Film-Tutorial von Synology: http://www.synology.com/enu/products/tutorial/webstation.php

i. Zugriffschutz eigener Seiten mit .htaccess

Was aber, wenn man nun einzelne Seiten der Allgemeinheit vorbehalten möchte und stattdessen per Passwort versucht abzusichern? Der einfachste Weg führt bei unserem Apache über eine .htaccess-Datei. Genau genommen müssen wir zwei Dateien erstellen. Zum einen die .htaccess und eine weitere, welche Nutzernamen und Passwörter enthält. Der Einfachheit halber, gehen wir davon aus, dass wir alle Seiten schützen möchten. Daher erstellen wir zunächst als Ablage für unsere Dateien einen Ordner „passwd“ im Stammordner „web“ des Apache. Nun erstellen wir mit einem einfachen Editor eine Datei mit folgendem Inhalt:

```
AuthName "Title"
AuthType "Basic"
AuthUserFile "/volume1/web/passwd/nutzer.pw"
require valid-user
```

Jetzt nennen wir diese Datei „.htaccess“ (Punkt nicht vergessen!). Wie wir sehen, wird im Code auf die Datei „nutzer.pw“ verwiesen. Dies ist die Datei, welche später die Zugangsdaten enthält. Also erstellen wir eine Datei nach folgendem Schema:

```
Nutzer:Passwort
```

Außerdem dürfen wir pro Zeile immer nur einen Nutzer einfügen. Da dieses System aber verhältnismäßig einfach zu knacken wäre, wird das Passwort stets verschlüsselt. Dazu gibt es im Internet eine ganze Reihe von .htaccess-Generatoren. Nun könnte unsere Datei zum Beispiel so aussehen:

```
Admin:$1$3a6ec4Pk$YiEMOVnuxlzRkQoEmLaPS0
Heinz:$1$vOb0WUyG$MnoANxLYwI2JWOSc96At5
```

Nun laden wir noch die Datei .htaccess in das Verzeichnis“ /web“ und die nutzer.pw in das Verzeichnis“ /web/passwd“. Wer allerdings nur Unterverzeichnisse sperren möchte, legt die .htaccess dort hinein.
Um die Sicherheit noch weiter zu erhöhen, kann man die nutzer.pw auch in ein anderes Verzeichnis als „web“ legen und erhöht somit noch einmal die Sicherheit.
Dieses kleine Beispiel schöpft bei weitem nicht die Möglichkeiten der .htaccess aus. Mehr Informationen findet man zu Hauf im Internet.

ii. Suchmaschinen abweisen

Suchmaschinen wie Google und Yahoo erleichtern uns das Leben im Internet jeden Tag gewaltig. Doch private Seiten sollten nicht unbedingt überall katalogisiert sein. Dafür gibt es „robots.txt“-Dateien. Die meisten Suchmaschinen suchen erst nach dieser Datei bevor sie mit ihrer Arbeit beginnen. Einfach eine Datei „robots.txt“ mit folgendem Inhalt in das Verzeichniss „/web“ des Apachen hochladen:

```
User-agent: *
Disallow: /
```

iii. Alle Zugriffe auf den Webserver loggen

Wer möchte nicht gerne mitlesen, wer die eigene Seite besichtigt. Mit einer kleinen Änderung im Apache zeichnet er alle IP-Adressen auf. Dafür müssen wir uns per SSH/Telnet auf unseren Server als „root“ einloggen. Als nächstes sichern wir die Datei, welche wir gleich bearbeiten werden:

```
cd /usr/syno/apache/conf
cp httpd.conf-user httpd.conf-user-backup
```

Nun öffnen wir unsere Datei:

```
vi httpd.conf-user
```

Und suchen den Eintrag „CustomLog /dev/null combined“ (normalerweise irgendwo zwischen Zeile 200 und 220). Jetzt muss er noch ersetzt werden:

```
CustomLog /volume1/web/log/access.log combined
```

Zwar warden jetzt alle IPs in der Datei „access.log” gespeichert, doch da das Verzeichniss „/web/log“ noch nicht existiert, würde er einen Fehler melden. Daher wechseln wir nun das Verzeichnis:

```
cd /volume1/web
```

… und erstellen „/log“:

```
mkdir log
```

Zu guter letzt müssen wir den Apache-Webserver noch neu starten:

```
/usr/syno/etc/rc.d/S97apache-user.sh restart
```

iv. Eigener “404-Error”

Die Überschrift ist eigentlich etwas irreführend. Denn der Begriff „404-Error“ wird bei der DiskStation nicht verwendet. Stattdessen bekommt man eine automatisierte Seite, welche besagt, dass die Seite nicht gefunden werden konnte. Dieser Fehler nennt sich allerdings offiziell „404-Error“. Wer nun eine eigene Fehlermeldung ausgeben möchte, erstellt zunächst eine html-Seite ganz nach dem eigenen Geschmack. Nun muss man die Datei in „missing.html“ umbenennen und in das Standardverzeichnis des Webservers „/web“ einfügen. Wer jetzt einfach mal eine nicht vorhandene Seite versucht zu öffnen, sieht seine neue Fehlermeldung (z.B. http://DiskStation/asdfghjkl/).

b. PhotoStation

Um die PhotoStation zu nutzen müssen wir diese erst aktivieren und dann die entsprechenden Ports weiterleiten um auch aus dem Internet darauf zugreifen zu können (http: 80 https: 443). Wenn wir uns jetzt auf der FileStation einloggen, sehen wir den neuen Ordner „photo“. Um Bilder oder Videos mit Hilfe der PhotoStation zu veröffentlichen, speichern wir diese dort. Nachdem wir Medien hinzugefügt haben, ist die DiskStation eine Weile beschäftigt, da es kleine Vorschaubilder von den Fotos genauso wie von den Videos anfertigt. Gibt es danach Probleme, z.B. das nicht alles angezeigt wird, hilft häufig der Knopf „Neu Ordnen“. Seit neueren Firmware-Versionen bietet die PhotoStation viele Möglichkeiten um das Aussehen anzupassen. Vorsichtig sollte man bei der Benutzerverwaltung

sein: Die PhotoStation besitzt eine eigene Datenbank. Nur der admin-Account läuft auch auf der PhotoStation. Neben der Bereitstellung von Medien bietet die PhotoStation auch einen Blog (Web-2.0-Begriff für Online-Tagebuch). Dieser arbeitet zwar nicht ganz so komfortabel und bietet nicht so viele Funktionen wie viele extern erhältliche Systeme, welche teilweise auch auf den DiskStations laufen, doch er verbraucht wesentlich weniger Ressourcen und ist direkt mit der Firmware verbunden.

c. FileStation

Wer nicht per ftp auf seine Daten zugreifen kann/will, dem steht die FileStation zur Verfügung um Zugriff auf alle freigegebenen Dateien zu haben. Ein Benutzeraccount ist für diesen Service allerdings Pflicht. Möchte man einen zusätzlichen Zugang schaffen, ohne dass Benutzer die Anmeldung des DiskStationManagers zu Gesicht bekommen, der kann über Port 6000 (http)/ 6001 (https) einen weiteren Zugang schaffen, welchen man auch individueller anpassen kann. Da Synology hier auf das http-Protokoll zurückgreift, kann man allerdings keine hohen Geschwindigkeiten wie bei ftp erreichen.

l. Zugriff über FTP

Die FileStation wird aber schnell unkomfortabel, insbesondere wenn man eine große Zahl von Dateien übertragen möchte. Wenn man daher häufiger Dateien überträgt, sollte man den FTP-Service aktivieren und nutzen. Um die Sicherheit zu erhöhen, sollte man allerdings immer mit der Option „Nur SSL/TLS-Verbindungen erlauben“ arbeiten. Desweiteren empfiehlt es sich, keinen Benutzer namens „guest“ oder „Gast“ zu betreiben und die Option „Anonymous FTP“ deaktivieren. Zum Zugriff über FTP empfiehlt es sich, ein spezielles Programm zu nutzen (z.B. WinSCP).

d. DownloadStation

Wer gerne größere Dateien aus dem Internet lädt und den eigenen Rechner nicht extra für längere Zeit laufen lassen will, kann mit der separaten DownloadStation Dateien vollkommen automatisch laden. Dabei werden auch verschiedene Protokolle wie zum Beispiel das Laden von einem Torrent unterstützt. Zum Nutzen der DownloadStation muss diese allerdings zunächst aktiviert werden und man benötigt einen Nutzer, welcher die DownloadStation nutzen darf (Untermenü „Anwendungen“, Benutzernamen und Passwort mit Manager/FileStation identisch).

e. AudioStation

Musik hören ohne einen Computer laufen zu lassen? Mit der AudioStation kein Problem mehr. Zunächst benötigt man dafür allerdings USB-Lautsprecher (kompatibilitäts-Liste beachten!), welche an einen der USB-Ports der DiskStation angeschlossen werden. Nun braucht man nur noch ein wenig Musik, welche in den neu erstellten Ordner „/music“ kopiert werden muss. Über das Webinterface kann man nun den Musikplayer steuern. Alternativ kann man auch die Synology-Fernbedienung kaufen und vollkommen ohne PC Musik abspielen. Auch IPods werden von der AudioStation unterstützt. Doch bisher haben wir uns nur die eine Seite der AudioStation angesehen. Denn wenn man die AudioStation über das Internet freigibt, kann man seine Musik mit Hilfe von Streaming auf irgendeinem Computer auf dieser Welt abspielen.

f. iTunes-Server

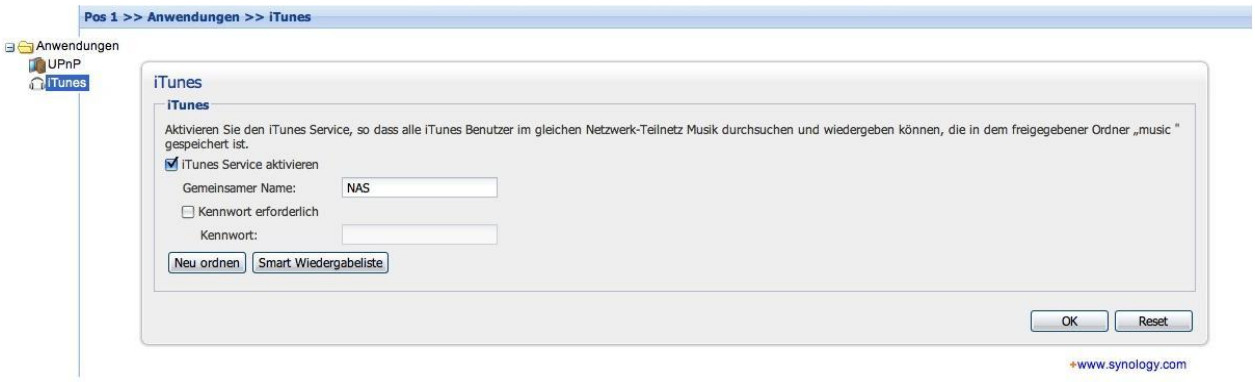
Was kann der iTunes-Server und was kann er nicht?

Der iTunes-Server ist als Streaming-Server konzipiert und ermöglicht allen Rechnern mit installierten iTunes den Zugriff auf die Musikinhalte des Ordners „music“. Er ist dabei als reiner Lieferer eines Musik-Streams gedacht, welcher mittels der in den Musikdateien enthaltenen „ID-Tags“ die Inhalte auf den Clients zur Anzeige bringt (bei fehlenden Album-Tag nimmt er den Ordnernamen als Tag). Er ist nicht zum Verwalten der Musik mittels einer Bibliothek oder ähnlichem gedacht. Das heißt auch, dass mit ihm kein synchronisieren mit einem IPod möglich ist!

Wie richte ich den itunes-Server ein?

Als erstes öffnet man das Management und aktiviert unter iTunes den entsprechenden Dienst. Hierbei wird automatisch ein Ordner „music“ erstellt, welcher mittels der Rechtevergabe in „Gemeinsame Ordner“ wie die anderen Ordner für den Zugriff zu konfigurieren ist.

„Gemeinsamer Name“ bezeichnet den Namen, unter dem die Musik später im iTunes zur Verfügung stehen wird.



Sollten Sie ausschließen wollen, dass jeder der Nutzer ihres Netzwerkes Zugriff auf die Musik erlangen soll, können Sie auch ein Passwort vergeben.

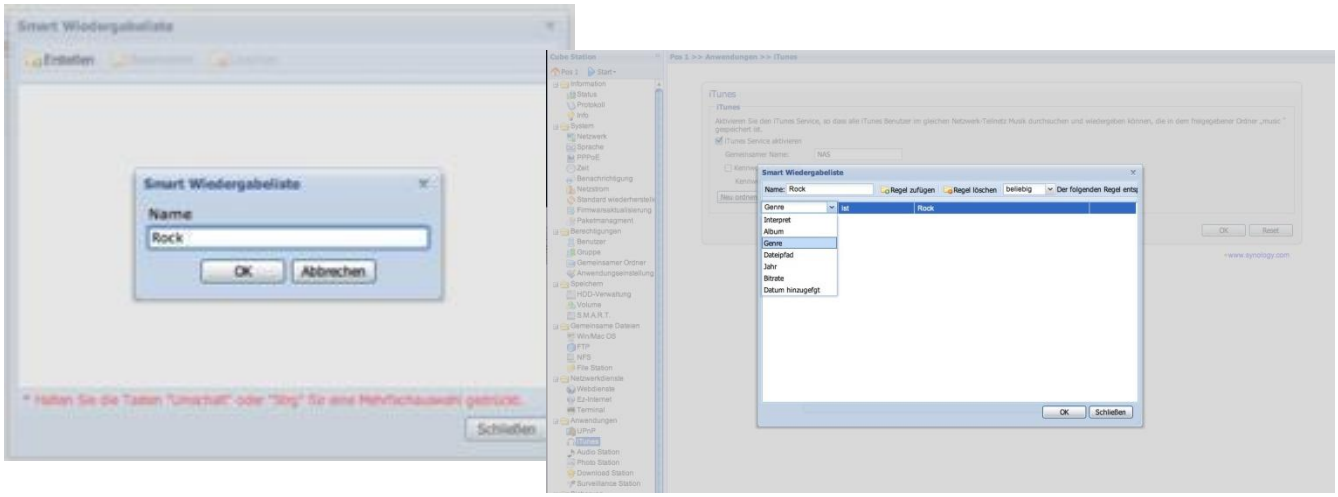
Sobald Musik in den Ordner gelegt wird, beginnt die DS mit der Indexierung entsprechend der in den Dateien vorhandenen Tags. Zu erkennen an der roten Schriftzeile:

Wenn Sie gerne bereits vordefinierte Filter nutzen wollen, können sie im Management eine passende „Smart Wiedergabeliste" erstellen, welche ständig innerhalb des Ordners „music" nach neuen Dateien Ausschau hält, welche eben die dort getroffenen Filter-Kriterien erfüllt.

Hier nun ein Beispiel:

Wir wollen alle Musik mit dem Genre Rock als gesonderte Liste angezeigt bekommen

Smart Wiedergabeliste auswählen-> Erstellen und Name vergeben, Regel hinzufügen und in dem folgenden Menü die Einstellung entsprechend vornehmen (Wenn Sie mehrere Kriterien auswählen wollen, einfach die nächste Zeile mit dem folgenden Kriterium bearbeiten) und mit OK bestätigen. Die folgende Frage sollten sie mit „Ja" beantworten und das Fenster schließen, wenn Sie keine weiteren neuen Filter einrichten wollen. Anschließend einmal „Neu Ordnen" auslösen, damit die Wiedergabeliste auch passend eingearbeitet wird.



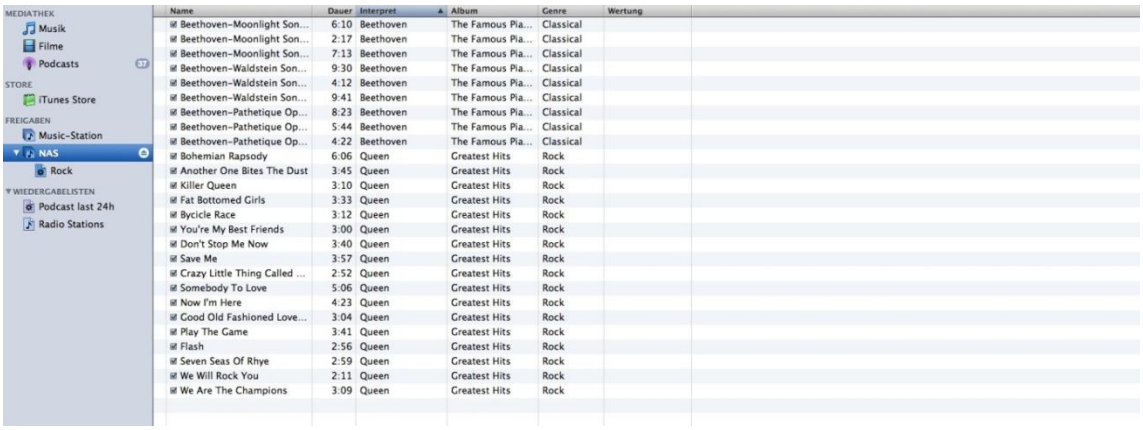
Allgemeiner Hinweis: Je mehr Smart-Wiedergabelisten erzeugt werden, umso mehr ist die DS mit der Filterung der Kriterien beschäftigt, welches längere Ladezeiten beim jeweils ersten Zugriff (einer jeden Nutzung) zur Folge hat, da diese Listen jedes Mal neu eingeladen werden müssen!

Damit haben wir auch schon die Funktionen des Servers abgearbeitet.

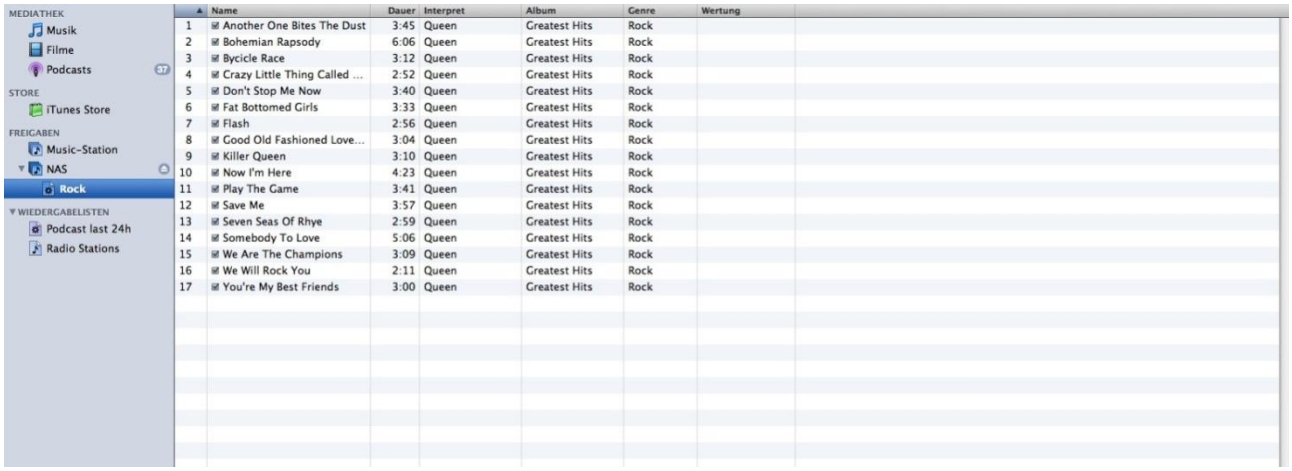
Kommen wir zum Client itunes:

Innerhalb iTunes unsere DiskStation als Freigabe angezeigt mit dem von Ihnen vorher gewählten Namen (hier NAS).

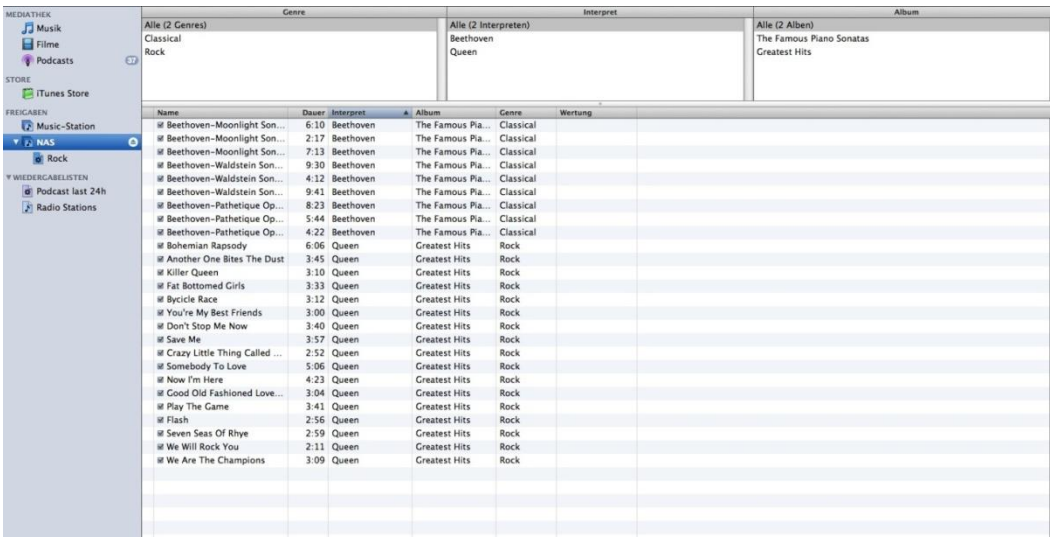
Wenn Sie nun auf diese Freigabe zugreifen, wird der Inhalt des Ordners „music“ nach dem Stand der Indexierung angezeigt. Durch klicken auf das, links daneben befindliche, Dreieck werden Ihnen die vorher erstellen Wiedergabelisten angezeigt.


MEDIATHEK
Musik
Filme
Podcasts
STORE
iTunes Store
FREIGABEN
Music-Station
NAS
Rock
WIEDERGABELISTEN
Podcast last 24h
Radio Stations
Name Dauer Interpret Album Genre Wertung
Beethoven–Moonlight Son... 6:10 Beethoven The Famous Pia... Classical
Beethoven–Moonlight Son... 2:17 Beethoven The Famous Pia... Classical
Beethoven–Moonlight Son... 7:13 Beethoven The Famous Pia... Classical
Beethoven–Waldstein Son... 9:30 Beethoven The Famous Pia... Classical
Beethoven–Waldstein Son... 4:12 Beethoven The Famous Pia... Classical
Beethoven–Waldstein Son... 9:41 Beethoven The Famous Pia... Classical
Beethoven–Pathetique Op... 8:23 Beethoven The Famous Pia... Classical
Beethoven–Pathetique Op... 5:44 Beethoven The Famous Pia... Classical
Beethoven–Pathetique Op... 4:22 Beethoven The Famous Pia... Classical
Bohemian Rapsody 6:06 Queen Greatest Hits Rock
Another One Bites The Dust 3:45 Queen Greatest Hits Rock
Killer Queen 3:10 Queen Greatest Hits Rock
Fat Bottomed Girls 3:33 Queen Greatest Hits Rock
Bycicle Race 3:12 Queen Greatest Hits Rock
You're My Best Friends 3:00 Queen Greatest Hits Rock
Don't Stop Me Now 3:40 Queen Greatest Hits Rock
Save Me 3:57 Queen Greatest Hits Rock
Crazy Little Thing Called ... 2:52 Queen Greatest Hits Rock
Somebody To Love 5:06 Queen Greatest Hits Rock
Now I'm Here 4:23 Queen Greatest Hits Rock
Good Old Fashioned Love... 3:04 Queen Greatest Hits Rock
Play The Game 3:41 Queen Greatest Hits Rock
Flash 2:56 Queen Greatest Hits Rock
Seven Seas Of Rhye 2:59 Queen Greatest Hits Rock
We Will Rock You 2:11 Queen Greatest Hits Rock
We Are The Champions 3:09 Queen Greatest Hits Rock

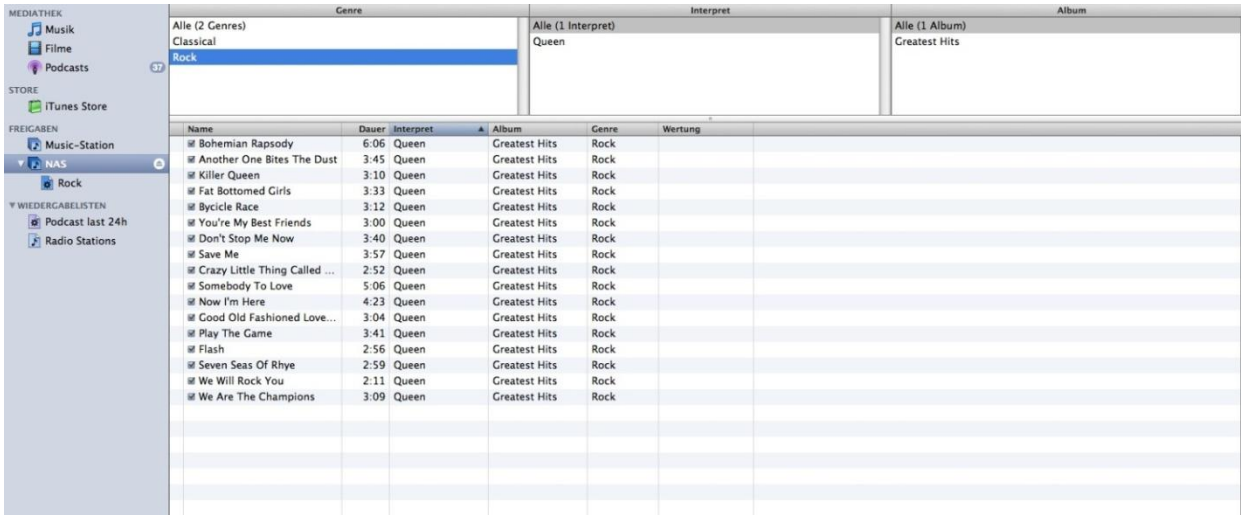
So sieht dann unsere fertige Wiedergabeliste aus:


MEDIATHEK
Musik
Filme
Podcasts
STORE
iTunes Store
FREIGABEN
Music-Station
NAS
Rock
WIEDERGABELISTEN
Podcast last 24h
Radio Stations
Name Dauer Interpret Album Genre Wertung
1 Another One Bites The Dust 3:45 Queen Greatest Hits Rock
2 Bohemian Rapsody 6:06 Queen Greatest Hits Rock
3 Bycicle Race 3:12 Queen Greatest Hits Rock
4 Crazy Little Thing Called ... 2:52 Queen Greatest Hits Rock
5 Don't Stop Me Now 3:40 Queen Greatest Hits Rock
6 Fat Bottomed Girls 3:33 Queen Greatest Hits Rock
7 Flash 2:56 Queen Greatest Hits Rock
8 Good Old Fashioned Love... 3:04 Queen Greatest Hits Rock
9 Killer Queen 3:10 Queen Greatest Hits Rock
10 Now I'm Here 4:23 Queen Greatest Hits Rock
11 Play The Game 3:41 Queen Greatest Hits Rock
12 Save Me 3:57 Queen Greatest Hits Rock
13 Seven Seas Of Rhye 2:59 Queen Greatest Hits Rock
14 Somebody To Love 5:06 Queen Greatest Hits Rock
15 We Are The Champions 3:09 Queen Greatest Hits Rock
16 We Will Rock You 2:11 Queen Greatest Hits Rock
17 You're My Best Friends 3:00 Queen Greatest Hits Rock

Als Alternative zum Anlegen von Wiedergabelisten sei Ihnen noch der Übersichtsfilter ans Herz gelegt, welcher durch anklicken bestimmter Filter sowohl beim Abspielen als auch bei der Suche Behilflich sein kann (STRG-B oder Apfel-B).


MEDIATHEK
Musik
Filme
Podcasts
STORE
iTunes Store
FREIGABEN
Music-Station
NAS
Rock
WIEDERGABELISTEN
Podcast last 24h
Radio Stations
Genre
Alle (2 Genres)
Classical
Rock
Interpret
Alle (2 Interpreten)
Beethoven
Queen
Album
Alle (2 Alben)
The Famous Piano Sonatas
Greatest Hits
Name Dauer Interpret Album Genre Wertung
Beethoven–Moonlight Son... 6:10 Beethoven The Famous Pia... Classical
Beethoven–Moonlight Son... 2:17 Beethoven The Famous Pia... Classical
Beethoven–Moonlight Son... 7:13 Beethoven The Famous Pia... Classical
Beethoven–Waldstein Son... 9:30 Beethoven The Famous Pia... Classical
Beethoven–Waldstein Son... 4:12 Beethoven The Famous Pia... Classical
Beethoven–Waldstein Son... 9:41 Beethoven The Famous Pia... Classical
Beethoven–Pathetique Op... 8:23 Beethoven The Famous Pia... Classical
Beethoven–Pathetique Op... 5:44 Beethoven The Famous Pia... Classical
Beethoven–Pathetique Op... 4:22 Beethoven The Famous Pia... Classical
Bohemian Rapsody 6:06 Queen Greatest Hits Rock
Another One Bites The Dust 3:45 Queen Greatest Hits Rock
Killer Queen 3:10 Queen Greatest Hits Rock
Fat Bottomed Girls 3:33 Queen Greatest Hits Rock
Bycicle Race 3:12 Queen Greatest Hits Rock
You're My Best Friends 3:00 Queen Greatest Hits Rock
Don't Stop Me Now 3:40 Queen Greatest Hits Rock
Save Me 3:57 Queen Greatest Hits Rock
Crazy Little Thing Called ... 2:52 Queen Greatest Hits Rock
Somebody To Love 5:06 Queen Greatest Hits Rock
Now I'm Here 4:23 Queen Greatest Hits Rock
Good Old Fashioned Love... 3:04 Queen Greatest Hits Rock
Play The Game 3:41 Queen Greatest Hits Rock
Flash 2:56 Queen Greatest Hits Rock
Seven Seas Of Rhye 2:59 Queen Greatest Hits Rock
We Will Rock You 2:11 Queen Greatest Hits Rock
We Are The Champions 3:09 Queen Greatest Hits Rock

Hier die entsprechende Auswahl „Rock“:


MEDIATHEK
Musik
Filme
Podcasts
STORE
iTunes Store
FREIGABEN
Music-Station
NAS
Rock
WIEDERGABELISTEN
Podcast last 24h
Radio Stations
Genre
Alle (2 Genres)
Classical
Rock
Interpret
Alle (1 Interpret)
Queen
Album
Alle (1 Album)
Greatest Hits
Name Dauer Interpret Album Genre Wertung
Bohemian Rapsody 6:06 Queen Greatest Hits Rock
Another One Bites The Dust 3:45 Queen Greatest Hits Rock
Killer Queen 3:10 Queen Greatest Hits Rock
Fat Bottomed Girls 3:33 Queen Greatest Hits Rock
Bycicle Race 3:12 Queen Greatest Hits Rock
You're My Best Friends 3:00 Queen Greatest Hits Rock
Don't Stop Me Now 3:40 Queen Greatest Hits Rock
Save Me 3:57 Queen Greatest Hits Rock
Crazy Little Thing Called ... 2:52 Queen Greatest Hits Rock
Somebody To Love 5:06 Queen Greatest Hits Rock
Now I'm Here 4:23 Queen Greatest Hits Rock
Good Old Fashioned Love... 3:04 Queen Greatest Hits Rock
Play The Game 3:41 Queen Greatest Hits Rock
Flash 2:56 Queen Greatest Hits Rock
Seven Seas Of Rhye 2:59 Queen Greatest Hits Rock
We Will Rock You 2:11 Queen Greatest Hits Rock
We Are The Champions 3:09 Queen Greatest Hits Rock

Allgemein empfiehlt es sich sehr, seine Musikdateien mit den richtigen Merkmalen (Interpret, Album, Genre, …) zu versehen, da nur so eine hohe Übersichtlichkeit bei der Nutzung garantiert ist!

Unterstützte Formate sind:
MP3, M4A, M4P

und ohne Anzeige von Merkmalen:
WAV, AIF

Ebenso werden Playlisten in den Ordnern unterstützt in folgenden Formaten:
M3U, WPL

Bevor ich dieses Thema schließe, noch kurz ein paar Worte zur iTunes-Alternative „Windows Media Player“ für „normale“ Nutzer. Dazu gibt es keinen offiziellen Dienst, doch das bereitstellen von Musik ist auch hier ohne Probleme möglich. Zunächst erstellen wir einen Ordner auf unserer DiskStation und vergeben die entsprechenden Rechte. Dann beladen wir ihn mit Musik und fügen ihn als Netzlaufwerk hinzu. Nun kann man den entsprechenden Ordner zur Medien-Bibliothek hinzufügen.

g. SurveillanceStation

Wer sein zu Hause auch von der Arbeit aus im Auge behalten möchte, oder einfach nur seiner DiskStation beim Betrieb zusehen mag, der kann eine IP-Kamera per USB anschließen und auf diese von überall zugreifen (Kompatibilität beachten!).

h. MailStation

Diese Funktion ist bei Synology als spk-Paket erhältlich und kann so einfach eingebunden werden.

Wer seine E-Mails immer Up-to-Date haben möchte, kann seine DiskStation dazu bringen, alle E-Mails automatisch abzuholen. Den eigenen Computer braucht man dann nur noch die Zugangsdaten der MailStation zu übergeben und kann schneller als gewohnt Mails empfangen und senden. Wer desweiteren keine Web-Oberfläche für seine Mails zur Verfügung hat, erhält mit der MailStation eine. Wer aus der MailStation einen vollwertigen Mail-Server machen möchte, sollte sich den nächsten Text anschauen:

i. MailStation als vollwertiger Mail-Server

Ein grundlegendes Problem ist, dass E-Mails, welche von dynamischen IP-Adressen empfangen werden, meist von dem Spam-Filtern abgefangen werden. Um dies zu verhindern müssen unsere Mails daher einen kleinen Umweg nehmen:

Zunächst erstellen wir uns einen **kostenpflichtigen** Account mit dynDNS bei selfhost.de Dieser bietet aber auch einen entscheidenden Vorteil: Ist die DiskStation aus welchem Grund auch immer offline, würde normal die Mail in den unweiten des Internets verschwinden. Selfhost.de wiederum speichert ihre Post, bis der Server sich zurück zur Arbeit meldet. Auf diese Weise geht garantiert nichts verloren.

Zuerst beginnen wir mit unserer Arbeit direkt an der DiskStation und wählen sie in der Administrationsoberfläche aus. Dort geben wir als Domainname die neue dynDNS-Adresse an. Bei allen anderen Optionen setzen wir einen Haken.

Nun loggen wir uns bei selfhost.de ein und legen einen neuen Mailspace über „Account -> Mail Admin“ an. Dort dann Spool anklicken und unsere DynDNS-Adresse auswählen. Jetzt noch die Daten, welche hinter „Username(fest)“ und „Passwort“ stehen notieren, damit wir sie später übernehmen können und zu guter letzt mit einem Klick auf „ändern“ bestätigen.

Jetzt geht die Arbeit an der DiskStation los. Wir öffnen unseren SSH/Telnet-Client (z.B. Putty) und beginnen:

```
cd /usr/syno/mailstation/etc
cp main.cf main-backup.cf
vi main.cf
```

In der gerade geöffneten Datei müssen wir jetzt bis zum Ende und fügen hinzu:

```
# selfhost
relayhost = [mail.selfhost.de]
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/usr/syno/mailstation/etc/smtp_auth
smtp_sasl_security_options = noanonymous

smtp_use_tls = yes
smtp_enforce_tls = yes
smtp_tls_enforce_peername = no

myhostname = euer Domain z.B. pax90.de
smtpd_sasl_auth_enable = yes
```

Nun öffnen wir eine neue Datei:

```
vi smtp_auth
```

Jetzt kommen unsere Zugangsdaten, welche wir auf den Zettel geschrieben hatten, zum Einsatz:

```
mail.selfhost.de USER:PW
z.B. mail.selfhost.de postmaster@masdasdoi.mail.selfhost.de:SDz78sJHD
```

Nun wandelt ihr die Datei in ein für die DiskStation lesbares Format um:

```
/usr/syno/mailstation/sbin/postmap smtp_auth
```

Damit wäre die Arbeit an der DiskStation beendet und wir können unseren neuen Mail-Server nutzen. Wer will, kann natürlich auch mit Outlook auf die Mails zugreifen. Für jeden Nutzer, der auf dem DiskStationManager erstellt wurden, existiert jetzt ein Mailkonto. Die POP3 und SMTP-Zugänge sind schlicht und einfach die DynDNS-Adresse.

i. Grundlegendes zum Thema Sicherheit

Hierbei handelt es sich zwar nicht um einen Bestandteil der Firmware, doch bevor wir im nächsten Kapitel die Backups behandeln, müssen wir uns erst einmal ein wenig die Sicherheit unserer Daten ansehen. Sie glauben mir nicht? Öffnen sie doch einmal die FTP-Ports und schauen sie nach ein paar Tagen in ihr Verbindungsprotokoll. Mittlerweile werden sie eine ganze Menge von Einträgen vorfinden, welche besagen dass jemand versucht hat ihr Passwort zu knacken. Meist wird dabei die „Brute-Force“-Methode verwendet. Mit anderen Worten: In möglich kurzer Zeit werden so viele Passwörter wie möglich ausprobiert. Vorausgesetzt dass der Benutzername stimmt, erhält der Hacker am Ende das richtige Passwort. Nur das ist genau was wir nicht möchten. Doch wir beginnen nicht an der DiskStation selber.

Zunächst beschäftigen wir uns mit dem Router. Diesen sollte man mit einem guten Passwort schützen und nicht vom Internet aus erreichbar machen. Darüberhinaus ist es wichtig, dass der Router eine eigene Firewall besitzt. Ist dies der Fall, sollte man einen anderen Router in Betracht ziehen. Als nächstes müssen wir damit beginnen Löcher in die Firewall zu schlagen. Wir sind bei den Ports angelangt. Hierbei gilt die simple Regel: So wenig wie nötig, so effizient wie möglich. Bevor sie sich jetzt die lange Port-Liste von Synology heranziehen und beginnen diese abzutippen, halten sie bitte inne. Zunächst überlegen sie, welche Dienste sie genau von überall nutzen möchten. Wenn sie die Download-Station nicht brauchen, sollten die entsprechenden Ports genauso unberührt lassen, wie wenn sie die AudioStation ausschließlich übers LAN betreiben. Als zweites schauen sie, ob es auch eine verschlüsselte Alternative gibt. So grenzt es an grobe Fahrlässigkeit die Ports für die Administrationsoberfläche per http zu öffnen. Wenn sie wirklich Zugang übers Internet brauchen, wählen sie lieber Port 5001 für https. Dasselbe gilt für alle anderen Dienste wie den Webserver und insbesondere Telnet. SSH hingegen können sie durchaus freigeben(Telnet 23, SSH 22). Nun sollte sich ihre Liste sehr verringert haben. Schließlich können sie nun mit dem abtippen in die Liste ihres Browsers beginnen. Für genaue Informationen zu diesem Thema konsultieren sie bitte das Handbuch ihres Routers, da sich dieser Prozess bei verschiedenen Herstellern sehr stark unterscheidet.

Eine kurze Bemerkung noch zu IDS bevor wir uns vom Router entfernen. Diesen Schutz bieten nicht viele Router. Haben sie jedoch die Möglichkeit, sollten sie diesen unbedingt nutzen. Dieser Dienst kann den Zugriff auf das Netzwerk als aller erstes stoppen bevor es überhaupt ihren NAS erreicht.

Wie bereits angekündigt lassen wir jetzt den Router hinter uns und wenden uns der DiskStation zu. Zunächst werfen wir einen Blick auf die Liste der Benutzer und ihrer Rechte. Die DiskStation macht das Verwalten dieser recht einfach und effektiv. Geben sie ihren Nutzern nur so wenig Rechte wie wirklich nötig. Insbesondere bei den Schreibrechten ist Vorsicht geboten. Besser sie müssen diese später auf Nachfrage des Anwenders nachreichen. Fordern sie ihre Nutzer außerdem dazu auf, sichere Passwörter zu verwenden (dazu später mehr).

Ich habe vorhin bereits Brute-Force-Attacken erklärt. Seit den letzten Firmware-Updates bringt die DiskStation eine eigene Sicherung gegen dieses Vorgehen mit. Dabei werden IP-Adressen, welche sich innerhalb von 10 Minuten mindestens 5mal fehlerhaft versucht haben einzuloggen, auf eine Liste gesetzt. Bevor sich ein Nutzer verbindet, wird diese Liste überprüft und bei Übereinstimmung die Verbindung abgebrochen, unabhängig davon ob die Zugangsdaten falsch oder richtig waren. Diese „IP-Blockierung“ muss zunächst aktiviert werden. Der FTP-Dienst bietet außerdem die Möglichkeit ausschließlich mit SSL/TLS verschlüsselte Verbindungen anzunehmen. Nutzt man FTP über das Internet, sollte man dies aktivieren. Beachten sie dann, dass ihr Programm, welches sie zum Zugang nutzen, dies unterstützt. (z.B. WinSCP)

Wie man den Webserver absichert haben wir bereits mit den „.htaccess“-Dateien besprochen. Wer in seinem Router die verschlüsselten Ports aktiviert hat, muss nun außerdem die entsprechenden Funktionen im DiskStationManager aktivieren.

Häufig wird auch nach Verschlüsselung der gesamten Festplatte gefragt. Ein grundlegendes Problem ist die hohe Prozessorleistung welche dafür benötigt wäre. Daher ist dies keine offizielle Funktion. „itari“ hat jedoch eine Möglichkeit gefunden, dies per Modifikation zu ändern (http://www.synology-forum.de/showthread.html?t=3343). Optional kann man natürlich auch einfach die Dateien bereits verschlüsselt auf der DiskStation abspeichern.

Zu guter letzt möchte ich noch ein paar Worte über Passwörter sagen. Hacker-Angriffe basieren häufig auf Wörterbüchern, welche häufig verwendete Wörter enthalten. Daher sollten sie nie Wörter in ihrem Passwort verwenden. Doch auch dies reicht bei weitem nicht aus. Ihr Passwort sollte nicht zu kurz sein (6-8 Zeichen MINIMUM!) und sowohl Zahlen als auch Buchstaben enthalten. Um die Sicherheit weiter zu erhöhen empfiehlt es sich außerdem, Sonderzeichen zu verwenden. Denn je länger ihr Passwort ist und je mehr Arten von Zeichen es verwendet, umso mehr Kombinationen muss ein Hacker ausprobieren. Ein Passwort welches alle diese Hinweise berücksichtigt (z.B. 12 Stellen mit Zahlen, Buchstaben und Sonderzeichen sowie kein erkennbares Wort) sollte es einem Angreifer sehr schwer machen. Bitter erinnern sie auch ihre Nutzer über diese Gefahr, denn am Ende sind sie auch für deren Daten verantwortlich, Passwort hin oder her!

Doch auch der beste Sicherheitsschutz kann einmal Löcher haben, daher sollten sie sich unser nächstes Thema gut durchlesen.

5. Backup

Über die Möglichkeit eines Raids haben wir ja bereits gesprochen. Doch was wenn man versehentlich alles formatiert (löscht) und dabei auch die wichtigen Firmen-Unterlagen mit verloren gehen? Oder wenn ein Hacker eindringt und entsprechendes veranlasst? Und wenn einmal mehrere Festplatten gleichzeitig kaputt gehen (oder bevor man eine neue beschaffen konnte)? Genug der Albträume. Denn hier geht es um deren Lösung. Man sollte regelmäßig ein Backup auf eine externe Quelle durchführen. Dabei kann es sich um eine externe Festplatte genauso handeln wie ein weiteres Synology-Produkt. Hier folgen kurze Beschreibungen aller möglichen Optionen.

a. Externe Festplatten-Sicherung

Die einfachste Möglichkeit die eigenen Daten zu schützen sind preiswerte, externe Festplatten. Diese können an freie USB-/eSATA-Schnittstellen gehängt werden. Zunächst sollte man sicherstellen, dass die neue Speichererweiterung erkannt wird. Nun zum DiskStationManager: Unter „Gerät -> USB-Laufwerk“ (oder eSATA-Laufwerk) formatieren sie die Festplatte. Hier haben sie die Wahl zwischen zwei Dateisystemen. Zum einen dem Linux-typischen „ext3“ und dem Windows-Standard „FAT32“. Am Ende ist die Entscheidung einem selbst überlassen. Doch sollte man bedenken dass Linux mit dem nativen Format, als ext3 schneller arbeiten kann und FAT32 darüberhinaus keine Dateien größer als 4 GB akzeptiert sowie keine Sonderzeichen als Dateinamen aufnehmen kann. Ist dies getan öffnen sie „Sicherung -> Lokale Datensicherung“. Nach einem Klick auf „erstellen“ öffnet sich ein Fenster, welches eine Schritt-für-Schritt-Anleitung enthält. Dazu gehören die Auswahl der zu sichernden Ordner und anderes.

b. Netzwerksicherung

Wer noch eine zweite DiskStation besitzt, kann diese zur Sicherung der eigenen Daten verwenden. Doch diese Möglichkeit ist nicht auf Synology-Produkte beschränkt. Synology verwendet das „rsync“-Protokoll zur Datensicherung, welches verschiedene Firmen unterstützen.

Zunächst muss dieser Dienst wie alle anderen aktiviert werden. Der entsprechende Kasten befindet sich unter „Sicherung -> Netzwerksicherungsdienst“. Anschließend richtet man die Sicherung unter „Sicherung -> Netzwerksicherung“ ein. Das Vorgehen ähnelt dabei dem der Sicherung auf eine externe Festplatte.

c. PC-Sicherung

Die Möglichkeit des Backups des eigenen PCs gibt es bei Synology auch. Dazu wird der „Data-Replicator“ verwendet, welcher auf der mitgelieferten CD vorliegt. Es empfiehlt sich aber sicherzustellen, dass man die aktuellste Version verwendet.

Nach der Installation ist es recht einfach möglich die Daten auf einer DiskStation zu sichern.

In der Vergangenheit hat sich allerdings gezeigt, dass der Data-Replicator nicht gerade ein ideales Tool zur Datensicherung ist. Zunächst sollte man sicherstellen, dass noch ausreichend Platz vorhanden ist. Denn der Data-Replicator legt einfach los, ohne dies zu überprüfen. Auch eine Verschlüsselung ist ihm fremd. Vor dem wiederherstellen der Daten wird nicht geprüft, ob die Sicherung noch intakt ist. Selbst nach erfolgreichem Wiederherstellen ist Vorsicht geboten, denn es kann vorkommen, dass Dateien welche bisher bestimmten Nutzern vorbehalten waren, plötzlich für alle verfügbar sind. Wer diese Funktion unbedingt benötigt und wichtige Daten sichert, sollte daher auf andere, eventuell kommerzielle, Programme setzen.

d. Einstellungen etc.

Bei eventuellem Reset des Gerätes ist es natürlich auch sehr hilfreich, wenn die Einstellungen sowie alle erstellten Benutzer(-Gruppen) mit gesichert sind. Wer die Netzwerksicherung verwendet hat, muss dies nicht separat erledigen. Sonst muss man allerdings eine spezielle Datei herunterladen. Die entsprechende Option findet sich unter „Sicherung -> Konfigurationssicherung“. Es empfiehlt sich außerdem alle Protokolle in regelmäßigen Abständen zu sichern um im Notfall Problemen auf die Spur kommen zu können.

6. Das Linux auf der DiskStation

a. Warum Linux?

i. Geschichte von Linux

Unsere kleine Exkursion beginnt 1983. Wer bereits ein wenig über die Ursprünge von Linux weiß, wundert sich nun, warum ich so früh beginne, denn zu diesem Zeitpunkt gab es noch nicht mal die Idee des Linux. Doch lange bevor Linux entstand das GNU-Projekt. Richard Stallman arbeitete zu dieser Zeit am MIT und wenn man damals Software bestellte, bekam man sie meist in Form des Quelltextes, damit man notfalls das Programm an den eigenen Computer anpassen konnte. Doch nun bekam es mehr und mehr Mode, die Software in binärer Form, also als direkte Dateien, auszuliefern um die eigenen Codes nicht zugänglich zu machen. Die Idee hinter GNU war die Schaffung eines freien, UNIX-ähnlichen Betriebssystems, welches mit allen nötigen Tools kommen sollte, um das System vom Kern an, für die eigenen Bedürfnisse und technischen Grundlagen anzupassen. Es dauerte mehr als ein Jahrzehnt, bis diese Grundbausteine fertig waren. GNU konnte viele frühe Erfolge einfahren, doch es hatte ein großes Problem: Es besaß keinen eigenen Kern, einen sogenannten Kernel. Stattdessen waren Nutzer immer noch darauf angewiesen, das „Halb-Betriebssystem“ auf einem bestehenden (kommerziellen) System wie UNIX zu installieren. Auch wenn sich viele nicht daran störten, wurden die Debatten um alternative Kernel immer intensiver, bevor Linux entstand. Auch wenn Linux nie ein richtiger Teil des GNU-Projekts wurde (Richard Stallman und sein Team arbeiteten stattdessen weiter am GNU HURD, dem ersten eigenen Kernel), basiert es trotzdem sichtbar auf GNU und setzt auch viele der dafür entwickelten Komponenten ein. Daher nennt man es auch „GNU/LINUX“.

Der Linux Kernel entstand mehr als ein Jahrzehnt, nach der ersten Fertigstellung des GNU-Projekts. Der finnische Student Linus Thorvalds (Universität von Helsinki) war unzufrieden mit den bisher entwickelten Kernel und begann daher die Arbeit an einem eigenen. Im Sommer 1991 schrieb er in einer Usenet-Newsgruppe einen kleinen Bericht über seine Arbeit und forderte die Leser auf, ihre Wünsche an ihn zu richten. So wuchs schnell die Zahl der Interessierten und Aktiven, da sich viele über diese Neuentwicklung freuten, welche nicht dieselben Kinderkrankheiten aufwies, wie die bisher entwickelten Lösungen. Schließlich wechselten sogar Entwickler der alternativen „Minix“ und „HURD“ -Systeme zum Linux-Projekt über, was dessen Gründer, Andrew Tanenbaum ziemlich aus der Fassung brachte.

Als Linux weiter wuchs, vergrößerte sich auch das Interesse an einer Lösung, Linux für Nicht-Entwickler zugängig zu machen, denn bisher musste Linux nach wie vor an das eigene System angepasst werden. Auch entwickelten sich Anwendungen, wie die Shell, welche wir nutzen um unsere DiskStations zu modden, zu grafischen Oberflächen weiter. So entstanden die ersten Distributionen. Diese besitzen eigene Installations-Routinen um den Nutzer durch diesen komplizierten Prozess zu begleiten und das System als Ergebnis selbst anpassen. Desweiteren bekommt man mit Distributionen fertige Anwendungen mitgeliefert, sodass man sofort loslegen kann.

ii. Linux unter GPL-Lizenz

Richard Stallman gründete im Zuge von GNU die „Free Software Foundation“. Eine nicht-kommerzielle Organisation, welche sich neben der Verwaltung der GNU-Entwicklung auch mit der Schaffung der GPL-Lizenz beschäftigte. GPL steht für „General Public License“. Ein großer Teil des Linux-Kernel steht unter dieser Lizenz. Software, welche unter GPL steht, kann von interessierten ohne Probleme geändert und angepasst werden, da der gesamte Quellcode zur Verfügung steht. Also könnte man zum Beispiel in GPL-Software einen Bug beheben und die neue Eigenentwicklung veröffentlichen. Doch die GPL besagt weiter, dass dieses neue Programm erneut frei zu Verfügung stehen muss.

iii. Linux ist frei verfügbar

Diese GPL-Lizenz macht es schließlich möglich, dass jedes Individuum und jede Firma ein eigenes Linux auf dem freien Kernel aufsetzen kann. So entstanden bereits Linux-basierte-Systeme für verschiedenste Geräte, wie Handys, Fernsehrekorder und Großrechner. So kommt es, dass auch Synology für die DiskStations auf Linux aufsetzt. Doch aufgrund der GPL ist der Code, auf welchem dieses Betriebssystem basiert, frei verfügbar. Für mehr Informationen lohnt sich ein Blick in den Supportbereich von Synology.

iv. Linux ist modular

Während Betriebssysteme wie Windows oder Mac als ganzes programmiert werden und später auch so laufen, besteht Linux aus vielen kleinen Einzelteilen, welche zusammen arbeiten um ein großes ganzes zu ergeben. Bei Linux wird sogar die kleinste Funktion in verschiedenste Mini-Programme aufgebrochen. Sogar die Lautstärke besteht aus verschiedenen Teilen. Was sich wie die unnötige Verschwendung von System-Ressourcen anhört, bietet bei genauem ansehen einen großen Vorteil: Bleibt einer dieser Bestandteile stecken oder erliegt einem Ausnahmefehler, kann er dies melden, ohne dass gleich das gesamte System betroffen ist und möglicherweise zusammenbricht. Außerdem macht es dies für Entwickler sehr viel einfacher, da der Quellcode viel übersichtlicher und einfacher zu warten ist.

vi. Linux ist sicher

Viele der bereits genannten Punkte spiegeln sich auch hier wieder. Denn wie soll man einen Virus programmieren, wenn man nie weiß, was das Zielsystem genau am Ende bereithält. Ein Virus, welcher für x86-Architektur gebaut wurde (x86 ist ein normaler Heimcomputer), hat unter x64, also bei 64-Bit-Systemen schlechte Karten. Wer sich hingegen als Angriffsstelle die Distribution aussucht, wird zwangsweise später genauso auf einer anderen Distribution landen, welche anders aufgebaut

ist. Auch die rege Community trägt ihren Teil dazu bei. Denn werden Probleme entdeckt, dauert es meist nicht lange, bis es entsprechende Lösungen gibt. Solche Sicherheitsrelevanten Themen werden meist in abgeschlossenen Chats diskutiert (schließlich sollte nicht jeder wissen, wo Probleme liegen). Wer nach noch einem weiteren Punkt schaut, stößt wieder auf den Modularen Aufbau. Denn wenn ein Virus ein Modul attackiert, ist die Wahrscheinlichkeit hoch, dass man am Ende nur einen winzigen Teil zerlegen konnte und das eigentliche System nach wie vor seinen Dienst verrichtet.

b. Die Verzeichnisse des Synology-Linux

Als erstes möchte ich auf den Verzeichnis-Aufbau des Synology-Linux eingehen. Dieser ist nur einsehbar, wenn man sich als „root“ über SSH einloggt. Die Firmware-eigenen-Funktionen geben keinen Blick auf die Stammverzeichnisse frei. Aber das zu Recht, denn die kleinsten Änderungen in diesem Teil des Linux können zu großen Problemen, bis hin zum Systemabsturz und Datenverlust, führen.

- /bin binaries – In diesem Verzeichnis werden alle Programme gespeichert, die vom Benutzer ausgeführt werden dürfen
- /dev devices – In diesem Ordner werden alle Treiber gepeichert.
- /etc et cetera – Alle wichtigen Konfigurationsdateien werden hier gespeichert.
- /etc.default Die originalen Konfigurationsdateien liegen hier als Backup vor.
- /initrd Enthält während des Bootens (Startphase) die Dateien, welche später zur „root-directory“, also „/“ wird. Das Stammverzeichnis, welches ungefähr mit „C:/“ bei Windows-Rechnern vergleichbar ist.
- /lib library – Alle wichtigen Standard-Funktionen der Programme, die häufig verwendet werden, sind in sogenannte Bibliotheken ausgelagert.
- /linuxrc rc=run commands – Datei mit automatisierten Startprogrammen. Dieser Ordner enthält eine Liste, welche Programme während der Startphase ausgeführt werden müssen.
- /lost+found Dieser Ordner wird vom fsck (file-system-check) genutzt, um beschädigte und/oder verwaiste Blöcke (Teile der Festplattenstruktur) zu speichern.
- /mnt mount point – Standard Verzeichnis, welches zum temporären „mounten“ genutzt wird. (mounten= verknüpfen zweier Dateisysteme unter Linux)
- /opt optional packages – Wird bei der Installation von ipkg angelegt und enthält dessen hinzugeladene Programme/Dateien.
- /proc Dieses Dateisystem, welches eigentlich gar nicht existiert, bietet eine Schnittstelle zu den Kernel- und Prozess-Informationen. Es enthält keine Dateien, stattdessen werden über Befehle wie „cat /proc/meminfo“ Informationen über den Status des Servers abgefragt.
- /root Standardverzeichnis des wichtigsten Benutzerkontos unter Linux (Systemadministrator).
- /sbin system binaries – In diesem Verzeichnis werden alle Programme gespeichert, welche besondere Privilegien erfordern (Systemverwaltung, Dienste, …)
- /sys Ähnlich wie /proc enthält es nur eine Schnittstelle zu den Kernel-Infos.

- /tmp temporary – Temporäre Dateien. Dieses Verzeichnis liegt nicht auf der Festplatte sondern im Arbeitsspeicher. Seine Kapazität ist meist rund 50% des Arbeitsspeichers. Genau wie bei Windows, können Teile ausgelagert werden um Platz für wichtigere Anwendungen zu machen.
- /usr user – Dies ist das Stammverzeichnis aller anderen Nutzer außer „root“.

c. IPKG

Als nächstes möchte ich das IPKG, auch bekannt als „Itsy Package Management System“, ein wenig betrachten. Um das Linux der DiskStation möglichst kompakt zu halten, besitzt das eingebaute Betriebssystem nur die notwendigen Komponenten und Bestandteile. Wer sich jetzt aber mit weiteren Funktionen befassen möchte, kommt kaum am IPKG vorbei. Ursprünglich wurde es für Geräte mit begrenztem Speicher wie Handys oder PDAs erschaffen. Mittlerweile hat es aber seinen Weg auf viele andere Plattformen geschafft. Die eigentliche Aufgabe ist das installieren und verwalten von Anwendungen auf Linux-Systemen. Wer beispielsweise openssh installieren möchte, tippt „ipkg install openssh“ in die Konsole (IPKG muss installiert sein). Hat openssh nun seine Arbeit getan und wird nicht mehr benötigt, oder man hat aus Versehen das falsche Programm installiert, kann mit „ipkg remove openssh“ wieder die Altlast loswerden. Um eine vollständige Liste mit allen unterstützten Anwendungen möchte, erhält diese mittels „ipkg list“. Da es sich um eine überwältigende Zahl von verschiedensten Anwendungen handelt, sollte man eher „ipkg list|grep irc“ nutzen. In diesem Beispiel erhält man eine Liste von allen Anwendungen, welche mit IRC (Internet Relay Chat) arbeiten.

i. Die Installation

Zunächst ist Vorsicht geboten, denn das „Management System“ unterscheidet sich von Prozessor zu Prozessor. Daher sollte man aufpassen, nicht das falsche Paket zu installieren. (Den Prozessor für das eigene Modell findet man bei Synology sowie in unserem Wiki.)

- Für ARM (armv5tejl): http://ipkg.nslu2-linux.org/feeds/optware/syno-x07/cross/unstable/syno-x07-bootstrap_1.2-5_arm.xsh
- Für PowerPC (ppc_6xx): http://ipkg.nslu2-linux.org/feeds/optware/ds101g/cross/unstable/ds101-bootstrap_1.0-4_powerpc.xsh
- Für PowerPC (ppc_85xx): http://ipkg.nslu2-linux.org/feeds/optware/syno-e500/cross/unstable/syno-e500-bootstrap_1.2-5_powerpc.xsh

Nun können wir mit der Installation beginnen. Zunächst müssen wir uns als „root“ über SSH einloggen. Nun muss man in das Verzeichnis /volume1/public wechseln, wo wir unser Paket zwischenspeichern wollen. Anschließend muss der Download gestartet werden. Hierbei unbedingt

auf den Prozessortyp achten! Dann muss noch das Skript gestartet werden und die Liste der zur Verfügung stehenden Pakete gelesen werden. Zu guter letzt natürlich noch die DiskStation neu starten.

```
cd /volume1/public
wget http://ipkg.nslu2-linux.org/... (Quelle für Prozessor einfügen)
sh syno-x07-bootstrap_1.2-5_arm.xsh
ipkg update
reboot
```

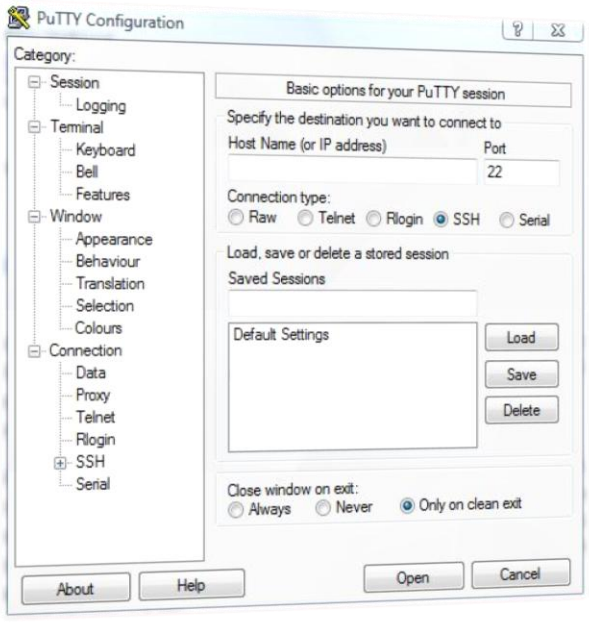
Zum Schluss kann die .xsh-Datei aus dem Ordner public gelöscht werden.

Hinweis: Wer den Ordner „public“ nicht nutzen will, kann einfach ein anderes Verzeichnis als Zwischenablage nutzen. Auch bei den Links sollte man vorsichtig sein, da die Datei durch eine neuere Version ersetzt werden kann. Dann sollte man einfach nachschauen ob es auf dem Server eine ähnliche Datei mit anderer Versionsnummer gibt.

d. Zugriff über SSH

Eigentlich müsste ich hier schreiben, Zugriff über SSH/Telnet. Doch Telnet ist eines der größten Sicherheitsrisiken beim Fernzugriff, daher möchte ich jeden dringendst dazu auffordern Telnet deaktiviert zu lassen. Durch die Verschlüsselung bei SSH gibt es keine Nachteile. Der einzige Unterschied liegt im Port. Doch dies sollte kein Problem darstellen.

Doch genug der Sicherheitsrisiken. Irgendwie lasse ich mich zu oft darüber aus. Sehen wir doch einmal was wir benötigen: Zuerst muss der Zugang aktiviert werden. Dann benötigen wir noch einen Client. Der berühmteste ist Putty. Dieser ist auch als „portable“-Version verfügbar, sodass man ihn auf einem USB-Stick mit sich herum tragen kann und bei Problemen auf der DiskStation schnell vom nächsten PC aus die Möglichkeit der Wartung hat. Die Konfigurations-Oberfläche sieht aus wie folgt (Screenshot: „Portable“-Version, v. 0.60):



Recht einfach erkennt man hier, dass ich noch keine Adresse eingegeben habe aber Port und Protokoll bereits auf SSH umgestellt sind. Wer nun noch die Adresse seiner DiskStation eingibt (entweder DynDNS-Account oder IP-Adresse), kann mit einem Klick auf „Open“ die Sitzung beginnen. Dann dauert es einen Moment bis eine Verbindung steht. Nun sitzen wir vor einem schwarzen Fenster auf dem ein grüner Cursor auf Befehle wartet. Zunächst will Putty die Anmelde-Daten von uns wissen. Man kann sich als „admin“ einloggen oder wenn man tiefer in das System möchte, auf den „root“ zurückgreifen. Sein Passwort ist normalerweise identisch mit dem des „admin“.

Was nun? Zunächst noch einmal der kurze Hinweis: SSH ist das wohl mächtigste Tool in der gesamten DiskStation. Ein falsches Kommando kann das gesamte System lahm legen. Ich weiß, ich erzähl zu viel darüber.

Wo sind wir eigentlich? Wer sich einen Überblick über das Verzeichnis machen möchte, in dem er gerade arbeitet kann dies mit „dir“ tun. Wem das Verzeichnis nicht gefällt, der kann dies mit „cd“ gefolgt vom Pfad ändern (cd=change directory). Auch das erstellen eines neuen Ordners gehört noch mit hier hin: „mkdir“ (make directory). Damit sollte man sich erst einmal recht komfortabel bewegen können. Natürlich ist das Wort „komfortabel“ eher vorsichtig zu gebrauchen, denn von unseren grafischen Oberflächen sind wir doch anderes gewohnt und tun uns ein wenig schwer mit dieser Variante.

Wer zum ersten Mal an SSH sitzt wird sich für andere Funktionen viel mehr interessieren, welche dem root vorbehalten sind: Das System-Log „more /var/log/messages“, das Start-Log „dmesg | more“, alle Prozesse „ps -ef“ und alle die derzeit aktiviert sind „top“, das beenden eines spezifischen Prozesses „killall [Programmname]“, sowie die derzeitige Auslastung der DiskStation mit „cat /proc/meminfo“ und „cat /proc/cpuinfo“.

Hinweis: Sämtliche in Klammern geschriebenen Worte müssen ersetzt werden (z.B. „killall httpd“).

Wer sich die Hilfe zu einem Befehl ansieht, dem fällt wahrscheinlich die lange Liste der Angaben auf, welche mit „-" beginnen. Dabei handelt es sich um Optionen, welche in beliebiger Reihenfolge und Menge nach dem eigentlichen Befehl gesetzt werden können.

e. Der/Die Apache-Webserver

Die Webserver? Ja richtig! Auf der DiskStation laufen mehr als ein einziger Webserver. Das Problem wäre, dass der DiskStationManager Zugriff auf sensible Daten benötigt. Hätte der normale Webserver derartige Berechtigungen, könnte das System sehr einfach infiltriert werden. Da der DSM aber prinzipiell auch „nur“ eine Website ist, benötigt es einen Webserver. Daher läuft neben dem normalen Webserver (/web), welcher unter dem Benutzer „nobody“ (engl. „Niemand“) werkelt, noch eine weitere Instanz, die gewöhnlich als System-Apache genannt wird. Diese nutzt die Berechtigungen des root.

Der gewöhnliche Apache, auch „Nutzer-Apache“ genannt, wird erst gestartet, wenn man einen der Prozesse aktiviert, welche gewöhnlich hier laufen: Den Blog, die PhotoStation und die WebStation, welche eigene Internetseiten bereitstellt.

Sämtliche Dienste welche über dieselbe Oberfläche laufen wie die Administration, also alles was über den Port 5000/5001 erreicht wird, werden auch dementsprechend vom System-Apache ausgeführt. Dazu zählen neben dem DSM die FileStation, die AudioStation und die DownloadStation.

7. CMS4DS

a. Wozu ein neues CMS?

Wer auf seiner DiskStation ein umfangreicheres Web-Projekt laufen lassen möchte, der wird mit sehr hoher Wahrscheinlichkeit schon einmal über sogenannte CMS-Systeme gestolpert sein. Ein „Content-Management-System“ verwaltet, wie der Name schon sagt, den gesamten Inhalt (Content) eines Web-Projektes selbstständig über eine eigene Verwaltungsseite. Gibt man nun mal bei Google „CMS“ ein, wird man hunderte von verschiedenen Systemen finden. Die Schwerpunkte liegen dabei sehr unterschiedlich. Während Wordpress beispielsweise ursprünglich als reine Blogsoftware entwickelt wurde, ist es zu einem der beliebtesten CMS und dem wahrscheinlich am weitest verbreiteten Blogsystem geworden. Joomla hingegen ist ein Universalwerkzeug um eine eigene Webpräsenz zu erstellen. Sowohl Wordpress als auch Joomla können durch eine Vielzahl von Plugins, welche von jeweils riesigen Communities entwickelt und bereitgestellt werden, mit fast allen nur denkbaren Funktionen ausgestattet werden.

Doch damit wäre immer noch nicht geklärt, warum itari und ein paar weitere Entwickler fleißig an einem eigenen CMS arbeiten. Eigentlich ist der Markt für derartige Software doch gesättigt, sollte man meinen. Doch nun zum Haken: Da die CMS eine Unmenge von verschiedensten Informationen sammeln und bereitstellen müssen, gibt es für die Entwickler meist nur eine Lösung: PHP und SQL. Wie wir bereits gesehen haben, muss SQL zunächst von PHP aufgerufen werden und das Ergebnis dann in html umgewandelt werden, bevor die Seite sich auf die Reise machen kann. Solange die CMS auf einem typischen Webserver laufen, also einer oft mehrere Tausend Euro teuren Hochleistungsmaschine, stellt das kein Problem dar. Nun besitzt unsere DiskStation aber weniger Rechenleistung als ein normaler Heim-PC (ist dafür aber sicherer, billiger, sparsamer und zuverlässiger). Aus Spaß oder zum Testen kann man gerne mal ein CMS auf der DiskStation installieren. Gerade wenn man kein „+“-Modell hat, sollte das Ergebnis allerdings nicht wirklich zufriedenstellend sein. Die Ladezeiten können dabei schnell viele Sekunden betragen. Es kann auch sein, dass von außen (Internet) aufgrund zu langer Ladezeiten die Anfrage abgebrochen wird und der Nutzer dann eine Fehlermeldung vor die Nase gesetzt bekommt.

Ich denke ich habe nun das Problem ausreichend dargestellt. Sehen wir einmal was itari dagegen unternommen hat.

b. Das Konzept

Wer jetzt dachte, er hätte ein tolles CMS für seinen kleinen weißen Wunderkasten gefunden, welches man einfach mit ein paar Klicks installieren kann, den muss ich leider enttäuschen. Itari möchte nicht nur ein schnelles CMS bereitstellen, sondern auch dass wir ein wenig über Web-Programmierung lernen.

So besitzt das CMS4DS weder eine fertige Installationsroutine, noch eine einfache Möglichkeit alles im Blick zu behalten, geschweige denn ein festes Design. Außerdem ist das Grundkonzept auch nicht

an eine Datenbank angebunden. Man muss sich daher ein wenig mit dem Aufbau einer Website sowie PHP und html auskennen. Auch weiterführende Kenntnisse zu SQL-Anbindung sind empfehlenswert, wenn man die Funktionalität ausbauen möchte. Ich werde trotzdem versuchen es so einfach wie möglich zu erklären, sodass man auch mit wenig Wissen über diese Gebiete etwas erreichen kann.

Bevor wir beginnen, noch ein wenig zu den Prinzipien hinter diesem CMS. Es sollte so wenig wie möglich PHP und anderen Server-seitigen Code besitzen. Gegen JavaScript und andere Client-basierte Sprachen spricht dagegen nichts. Wer will kann mit Java ein Feuerwerk von Effekten auslösen, solange er so wenig wie möglich PHP dabei einsetzt. So können wir sicherstellen, dass unser cms4ds das bleibt, wofür es geschaffen wurde: schnell. Desweiteren ist bei Browser-Weichen Vorsicht geboten. Also bei Code, der von Browsern unterschiedlich interpretiert wird oder den manche gleich gar nicht verstehen. Die Grundlagen wie wir sie hier besprechen werden, laufen mindestens mit Firefox und Internet Explorer ohne Probleme.

Trotz des Namens ist unser cms4ds am Ende halt „nur“ eine umfangreiche Internetseite. Daher läuft es nicht nur auf einer DS sondern auf jedem Server, der PHP5 und eventuell eine Datenbank bereitstellt. Das trifft meist auch auf bezahlten Webspace zu. Bei kostenlosen Hostern ist es aber eher die Ausnahme. Da es keine weiteren Änderungen am Linux benötigt, ist es außerdem keine 3rd-Party-Application sondern eine einfache Webanwendung, also muss man von Seiten Synology nichts befürchten.

c. Der Aufbau

Nun geht es ans Eingemachte. html-Dateien besitzen im cms4ds nur eine kleinere Rolle, sie besitzen die Verweise zu den verschiedenen Bestandteilen wie dem header und footer (Kopfelemente und Fußleiste). Der wichtigste Teil gestaltet sich dabei wie folgt:

```
<!--#include virtual="header.html" -->
<!--#include virtual="cms4ds.php?where=cms4ds" -->
<!--#include virtual="footer.html" -->
<script>LoadRSS2("http://www.synology-forum.de/external.html?count=12");</script>
<!--#include virtual="bottom.html" -->
```

Schritt für Schritt:

In der ersten Zeile wird der header, also die Überschrift und das kleine Bild sowie das Menü am oberen Rand, eingebaut.

Die zweite Zeile lädt eine PHP-Datei, welche den gesamten dynamischen Content lädt. Über die Angabe „?where=“ wird die Position per Variable übergeben. Diese Variable ist von Seite zu Seite unterschiedlich.

Die dritte Zeile enthält den sogenannten footer. Er enthält den Hinweis des Copyrights sowie ein paar kleine Zahlen, wie lange der Aufruf gedauert hat.

Als nächstes wird der blaue Kasten am rechten oberen Rand gefüllt. Itari nutzt dabei einen von unserem Forum bereitgestellten RSS-Feed.

Die letzte Zeile, die bottom.html, lädt im Hintergrund alle Unterseiten vor, zeigt zunächst aber nur die erste an. Alle anderen zeigt es in einer kleinen Liste rechts unten an.

d. Die html-Seiten im Detail

Jetzt kommen wir zur genauen Funktion der einzelnen Dateien, von denen wir bisher nur gesehen haben was sie machen sollen. Nun mal einen Blick darauf, was sie wirklich machen.

i. Header.html

Oben haben wir gesagt, dass der header sich mit der Überschrift, dem Kopfbild und dem Menü befasst.

```
<!--#set var="t" value="" -->
<html><head><title>cms4ds</title>
<link rel="stylesheet" type="text/css" href="cms4ds.css" />
<link rel="alternate"  type="application/rss+xml"
                       title="cms4ds-News-Feed" href="cms4ds_rss.php"/>
<link rel="shortcut icon" href="cms4ds.ico" />
<link rel="search" type="application/opensearchdescription+xml"
      href="http://syno/cms4ds/search.xml" title="cms4ds" />
<script type="text/javascript" src="cms4ds.js"></script>
<script>var d0,d1,d2,d3;d0=new Date().getTime();</script>
</head><body onload="history()">
<iframe src="blank.html?0" name="histFrame" id="histFrame"></iframe>
<div id="basecontainer">
<div id="top">cms4ds<br/><span>a CMS for Synology Disk Stations</span></div>
<div id="menu">
<a href="cms4ds.html">cms4ds</a>
<a href="files.html">files</a>
<a href="stylesheet.html">stylesheet</a>
<a href="javascript.html">javascript</a>
<a href="database.html">database</a>
<a href="rss.html">rss</a>
<a href="search.html">search</a>
<a href="tools.html">tools</a>
</div> <!-- menu -->
<div id="container">
<div id="container_col">
<input id="search" ondblclick="searchAll(this.value)"
       title="Doppelklick zum Starten des Suchvorgangs"
       value="(Suchbegriff)"/>
<div id="counter"># Besucher</div>
<div id="feed"></div>
<div id="index"></div>
</div> <!-- container_col -->
<div id="content">
```

Genau genommen macht er aber noch viel mehr. Im ersten Teil wird die CSS-Datei, welche das Aussehen vorgibt und der RSS-Feed, verlinkt. Außerdem wird der Dateiname des kleinen Mini-Bildes für den Browser definiert. Außerdem bieten ja bekanntlich verschieden Browser kleine Suchleisten an, die einem das Eintippen der Suchseite erspart. Damit auch unsere Seite entsprechend genutzt werden kann, wird die xml-Datei verlinkt, welche dafür verantwortlich ist.

Die nächsten Zeilen sind besonders wichtig. Die größten und wichtigsten Funktionen werden nämlich per JavaScript abgewickelt, da sie unser CMS schlank halten. Der entsprechende Link befindet sich in diesen Zeilen. Was genau dort passiert, werden wir später sehen.

Mit dem Beginn des eigentlich sichtbaren Teils (<body>) wird die Funktion „history“ ins Leben gerufen. Das ganze läuft über einen iframe. Mit einer iframe kann man einen gewissen Bereich von einer anderen html-Datei einbinden. Das einzig wirklich wichtige für uns ist, dass die „blank.html“ vorhanden ist, damit wir nicht das Risiko eines Fehlers bei manchen Browsern eingehen.

Um den Rest des Textes ein wenig kurz zu halten: Es wird die der Kopf mit Überschrift und Bild sowie das Menü eingefügt. Um für das Menü nicht noch gleich ein paar mehr Skripte zu verbrauchen, werden gleich die Links mit eingefügt (die lange Liste <a href…). Außerdem wird die Struktur des rechten Kastens vorbereitet. Der Inhalt kommt, wie fast überall, von einer externen Quelle.

Nur der Vollständigkeit halber: Alle weiteren nötigen Informationen kommen von einer CSS-Datei. Itaris Hauptgedanke dahinter ist, Inhalt und Design so klar wie möglich zu trennen.

ii. footer.html

```
</div> <!-- content -->
</div> <!-- container -->
<div id="footer">(c) 2008 by &#x81F3; itari based on cms4ds</div>
<div id="timer"><!--#echo var="t" --> # </div>
</div> <!-- basecontainer -->
```

Der „footer“ enthält die untersten zwei Zeilen des CMS. Die erste beinhaltet den Copyright-Hinweis (Quelltext Zeile 3) und die zweite die gemessenen Zeiten, wie lange es gedauert hat um die Seite zu laden.

iii. bottom.html

```
<script>
make_index();
s=window.location.search; if (s.substr(1,1)=='s') searchAll(s.substr(3));
if (s) showme(s);
</script>
</body></html>
```

Obwohl diese Datei kurz ist, stellt sie einen wichtigen Teil da. Auch für den reinen Anwender wichtig zu wissen, ist, dass die Funktion „make_index“ alle **Überschriften mit h3-Markierung** heraussucht und dann als **Überschriften für die Beiträge**, deren Liste rechts unten angezeigt wird, festlegt. Außerdem wird die Eigenschaft „display“ aller anderen Beiträge wird auf „none“ gesetzt, damit sie unsichtbar werden und erst dann erscheinen wenn man sie gerne hätte.

Noch als letzter Hinweis: Man sollte darauf achten, dass alle Ordner mit .htaccess ausreichend abgesichert sind! Der Download von itari bringt schon ein paar mit, doch die treffen nicht für jeden Einsatzzweck zu. Außerdem schalten sie die SSI (ServerSideIncludes), wie PHP und andere Funktionen, ein.

e. Stylesheet

Mit den html-Dateien haben wir zwar die Grundsteine gelegt um Inhalt wiederzugeben, doch bisher fehlt jegliche Aufteilung der Elemente, eine Definierung der Schriftart, Farben, …

Ich denke nicht, dass es nötig ist jeden einzelnen Teil der cms4ds.css zu erklären. Wer dies möchte, sollte auf itaris Seite vorbei schauen (http://itari.syno-ds.de). Stattdessen werde ich nur schnell erklären, wie man das Design nach den eigenen Belieben anpassen kann.

Nehmen wir als Beispiel diese Zeile:

```
h4          {font:13 Georgia, Serif;color:#69c;}
```

Jede einzelne der Zeilen in der CSS beginnt mit der Angabe, auf welchen Bereich es zutrifft. Hier handelt es sich dabei um alle, mit h4 formatierten, Überschriften. Es folgt eine Klammer „{“ und die erste Deklaration. Zunächst geben wir an, was genau wir ändern möchten. Hier handelt es sich mit dem Begriff „font“ um Schriftgröße und Schriftart. Hinter dem Doppelpunkt folgt dann, wozu es geändert werden soll. Bei obigem Beispiel also zu einer Schrifthöhe von 13 Pixel und einer neuen Schriftart. Es wird empfohlen, mehrere Schriftarten oder am Ende eine Schrift-Familie anzugeben. In unserem Beispiel ist die Standard-Schriftart „Georgia“. Ist diese, aus welchem Grund auch immer, nicht vorhanden, wird auf den nächsten Begriff in der Reihenfolge zurückgegriffen. „Serif“ ist keine Schriftart, sondern eine Schriftartenfamilie. Ist „Georgia“ also nicht vorhanden, weicht der Browser auf eine andere Schriftart aus, welche zu dieser Gruppe gehört.

Nachdem ein Semikolon gesetzt wurde, geht es weiter. „Color“ meint natürlich die Farbe welche unsere h4-Überschrift später haben wird. Es gibt zwei Wege, Farben zu definieren, welche von jedem aktuellen Browser korrekt interpretiert werden.

Die erste nutzt für die 16 Grundfarben einfache Namen wie „black“ oder „red“. Die Netscape-Entwickler wollten dies noch weiterführen und haben daher noch mehr Möglichkeiten in ihren Browser eingebaut. Diese sind, im Gegensatz zu den 16 wichtigen, nicht Teil des html-Standards und können daher falsch interpretiert werden.

Um genaue Farben anzugeben, gibt es daher noch eine andere Methode. Sogenannte „RGB-Werte“ verwandeln Farben in einen recht simplen, sechsstelligen Zahlen- und Buchstabencode. Beginnend mit einer Raute („#“) werden dabei die jeweiligen Anteile von Rot, Grün und Blau angegeben. Für jede Stelle gibt es 16 Möglichkeiten. Zunächst die Zahlen von 0 bis 9 und für 10-15 werden die Buchstaben A-F verwendet. So ergeben sich am Ende 256 Farben welche per html-Code übertragen werden können.

Soviel zu einer kleinen Einführung in das CSS welches itari verwendet.

f. Javascript

Ganz am Anfang habe ich geschrieben, was das cms4ds so besonders macht. Seine Geschwindigkeit. Und warum? Weil itari sehr sparsam mit serverseitigen Programmiersprachen wie PHP ist. Stattdessen ist das Herz des CMS4DS fast ausschließlich JavaScript.

Also stürzen wir uns nun auf die cms4ds.js. Wie die Dateiendung „.js“ schon verrät, enthält diese Datei keine einzige Zeile PHP oder html sondern ausschließlich Javascript. Sie ist wohl eine der längsten und kompliziertesten Dateien im gesamten Verbund. Umso wichtiger ist es, dass wir nun Schritt für Schritt uns die wichtigsten Funktionen ansehen.

```
var isie = /msie/i.test(navigator.userAgent) && !window.opera;
var myXMLHTTPRequest = (window.XMLHttpRequest)?
                        new XMLHttpRequest():
                        new ActiveXObject("Microsoft.XMLHTTP");
var d0,d1=0,d2=0,d3; d0=new Date().getTime();
if (!document.getElementsByClassName) {
          document.getElementsByClassName = function(className) {
                    var children = document.getElementsByTagName('*') || document.all;
                    var elements = new Array();
                    for (var i = 0; i < children.length; i++) {
                              var child = children[i];
                              var classNames = child.className.split(' ');
                              for (var j = 0; j < classNames.length; j++) {
                                        if (classNames[j] == className) {
                                                  elements.push(child);
                                                  break;
                                        }
                              }
                    }
                    return elements;
          }
}
function $(e) { return document.getElementById(e); }
function $$(e){ return document.getElementsByClassName(e); }
```

Die erste Zeile überprüft den Browser. Die nächsten legen ein paar Variablen an, die insbesondere für die Zeitmessung am unteren Ende der Seite benötigt werden. Der lange mittlere Teil beschreibt die neue Funktion document.getElementsByClassName. Diese macht es möglich auf ein html-Objekt statt den Namen des Objekts, den Namen dessen Klasse zu verwenden.

In der letzten Zeile wird dann eine weitere Funktion hinzugefügt, die eigentlich nicht viel mehr macht, als den Namen zu kürzen. (Ja, Entwickler sind faul)

Die Zeile davor liefert statt den Klassennamen, die ID zurück, macht aber eigentlich auch nicht viel anderes als die ihm folgende Zeile, denn sie kürzt nur den Namen einer bereits vorhandenen Funktion.

```
function LoadHTML(request){
d1=new Date().getTime();
  myXMLHTTPRequest.open("GET", request, false); myXMLHTTPRequest.send(null);
  $('content').innerHTML=myXMLHTTPRequest.responseText;
d2=new Date().getTime();
}
function LoadHTML2(request){
d1=new Date().getTime();
  myXMLHTTPRequest.open("GET", request, false); myXMLHTTPRequest.send(null);
  $('index').innerHTML=myXMLHTTPRequest.responseText;
d2=new Date().getTime();
}
```

Diese beiden Funktionen werden erst später wirklich wichtig, wenn wir uns dem RSS-Feed nähern. Denn diese Funktionen machen es möglich, dass der Feed den Titel und den Inhalt abzufragen.

```
var counter;
var ref=0;
function show(i){
  $$('content')[ref].style.display="none";
  $$('content')[i].style.display="block";
  histFrame.location.search=i;
}
function make_index() {
 var l=$('content').childNodes.length;
 var i=0;
 var j=0;
 var out="";
while(i<l) {
  if( $('content').childNodes[i].firstChild != null) {
    out+="<div class='index' onclick='show("+j+")'>"
       +$('content').childNodes[i].getElementsByTagName('H3')[0].innerHTML+"</div>";
    $$('content')[j].style.display='none';
    ++j;
    }
  ++i;
  }
myXMLHTTPRequest.open("GET", "cms4ds_log.php?loc="
  +location.pathname+location.search, false); myXMLHTTPRequest.send(null);
$('counter').innerHTML=
  $('counter').innerHTML.replace('#',myXMLHTTPRequest.responseText);
$('index').innerHTML=out;
$$('content')[0].style.display='block';
d3=new Date().getTime();
$('timer').innerHTML=$('timer').innerHTML.replace('#','AJAX-Request: '
  +((d2-d1)/1000)+' sek. | Page: '+((d3-d0)/1000)+' sek.');
}
```

Nachdem ein paar weitere Variablen definiert wurden, folgt die Funktion show(). Diese kümmert sich um das Beitrags-Menü rechts unten und schaltet zwischen den Beiträgen hin und her. Was dem Nutzer verborgen bleibt, ist dass die URL geändert wird um den „zurück“-Knopf am Browser eine Chance zu geben, das richtige zu tun.

Wie bereits beschrieben, werden die Beiträge anhand der h3-Überschriften unterteilt und mit Namen verpasst. Die Funktion make_index() erledigt genau dies. Sie läuft den gesamten Inhalt der Seite ab und richtet die Navigation für die Beiträge entsprechend ein.

Ist der Index fertig und die Seite geladen, wird die cms4ds_log.php aufgerufen und mittels PHP wird der Seitenaufruf geloggt um später jeden Aufruf minuziös verfolgen zu können und die Zahl der Benutzer wird aktualisiert (um eins erhöht).

```
function showme(s) {
  var found=0, page='', id='';
  s=decodeURIComponent(s.substr(1)); sa=s.split('&');
  for(var i=0; i< 0) location.pathname=encodeURI('cms4ds/'+page+'.html?id='+id);
  for(var i=0; i< $$('index').length; ++i)
    if (id == $$('index')[i].innerHTML) {
```

```
      $$('content')[i].style.display="block"; ref=i; found=1;
      histFrame.location.search=i;
      }
    else
      $$('content')[i].style.display="none";
  if (found == 0) {
      $$('content')[ref].style.display="block";
      }
}
```

Die Funktion showme() ist ein Teil der Suche. Die Funktion selber ist nicht in der Lage, eigenständig die Beiträge zu durchforsten. Stattdessen nimmt es das Ergebnis entgegen und ruft dieses auf. Die eigentliche Suche übergibt die Rubrik und die ID des Artikels. **Daher funktioniert die Suche nur zuverlässig wenn alle Rubriken unterschiedliche Namen haben!** Wenn irgendetwas falsch gelaufen ist und showme() kein Ergebnis finden kann, leitet es wieder zur letzten Seite zurück.

```
function searchAll(s) {
$('content').innerHTML='';
if (s!='') {
  d0=new Date().getTime();
  LoadHTML("cms4ds.php?like="+s);
  for (var i=0; i<$('content').childNodes.length; ++i)
    if ($('content').childNodes[i].nodeType == 1)
      $('content').childNodes[i].innerHTML
        =$('content').childNodes[i].innerHTML.replace(eval('/('+s+')/gi'),'<u>$1</u>')
  }
if ($('content').innerHTML=='')
  $('content').innerHTML='<div><h3>keinen Suchbegriff gefunden</h3></div>';
make_index();
}
```

Nun zur eigentlichen Suche. SearchAll() ruft mit Hilfe von AJAX die Datenbank auf und sucht nach Übereinstimmungen. Nachdem am Anfang der „content“, also der Mittelteil der Seite, geleert wurde, wird es jetzt mit dem Ergebnis gefüllt.

```
function bookmark(s) {
  url=window.location.href+'?id='+s;
  if (isie) window.external.AddFavorite(url,s); else window.sidebar.addPanel(s,url,'');
}
```

Die Funktion bookmark() stellt sicher dass eine Seite vom Browser vernünftig als Bookmark/Favorit gespeichert werden kann. Da es hier Unterschiede zwischen den Browsern gibt, wird die Funktion entsprechend abgesichert.

```
function history(){
  if (histFrame.location.search) {
    his = histFrame.location.search.substr(1);    // remove '?'
    if (his != ref) {                             // poll return on no change
      $$('content')[ref].style.display="none";
```

```
      $$('content')[his].style.display="block";
      ref=his;
      }
    }
  window.setTimeout(history,1000);                // Timer
}
```

Damit wären wir schon fast am Ende angelangt. Das Problem der History des Browsers liegt darin, dass alle Beiträge vollständig geladen werden und er somit nicht zwischen ihnen unterscheidet. Mit einem Klick auf „Zurück“ im Browser würde man daher einen zu großen Sprung machen. Um dies zu verhindern, hat itari an eine eigene Funktion gedacht, die dem Browser die Arbeit abnimmt und zielsicher zum vorherigen Beitrag leitet. Die Funktion wird jede Sekunde aufgerufen. In der letzten Zeile steht zwar eintausend, doch Javascript arbeitet wie die meisten Programmiersprachen in Millisekunden und muss daher mit dem Wert „1000“ versorgt werden. Es prüft dann, ob im „histFrame“ unter „location.search“ eine Änderung stattgefunden hat. Trifft dies zu, wechselt sich der Beitrag.

Damit hätten wir uns durch das Herz von itaris CMS4DS gekämpft. Von nun an, ist das CMS prinzipiell zur Arbeit bereit. Doch itari hat natürlich nicht hier aufgehört. Uns steht noch eine Tour durch die zusätzlichen Funktionen wie dem RSS-Feed und der Datenbankanbindung bevor.

f. CMS4DS+SQL=?

Wie die Überschrift schon symbolisiert: Wir stürzen uns jetzt auf die Möglichkeit, eine SQL-Datenbank mit dem CMS4DS zu verknüpfen. Die Datenbank muss über ein geeignetes Admin-Programm zunächst erstellt werden.

Zunächst sehen wir uns einmal an, in welcher Form die Datenbank aufgebaut werden soll:

```
CREATE TABLE cms4ds_content (
  Content_ID int(11) NOT NULL auto_increment,
  TS timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
  Titel varchar(200) collate latin1_general_ci NOT NULL,
  Titel_Datum datetime default NULL,
  Hide_Date char(1) collate latin1_general_ci default NULL,
  Content varchar(32000) collate latin1_general_ci default NULL,
  Rubrik varchar(200) collate latin1_general_ci default NULL,
  Sequence int(11) default NULL,
  PRIMARY KEY  (Content_ID)
)
```

Content_ID: Existiert ausschließlich zur besseren Handhabung.

TS: Enthält das Datum der letzten Änderung, wird nicht direkt vom CMS genutzt.

Titel: Enthält eine Zeichenfolge, welche dem Namen des Beitrages entspricht.

Titel_Datum: Hier wird das Datum der Veröffentlichung vermerkt. Das Datum dient außerdem als sekundäres Suchkriterium, welches dazu genutzt wird um die Artikel zu ordnen. (Nach dem Titel)

Hide_Date: Ist hier der Wert „y“ eingefügt, wird das Datum in der Ausgabe versteckt.

Content: Enthält den eigentlichen Beitrag.

Rubrik: Gibt an, zu welcher Rubrik der Beitrag gehört. Es können auch mehrere Werte, mit Komma getrennt, angegeben werden.

Sequence: Wenn benötigt, kann hier eine manuelle Reihenfolge für die Sortierung der Beiträge in Form einer Zahl angeben. Es wird dabei aufsteigend sortiert.

Nun müssen wir noch dafür sorgen, dass ein Zugriff auf die Datenbank erfolgen kann. Dazu wird die Datei „cms4ds.php.inc“ verwendet. Die anderen Dateien enthalten einen entsprechenden Aufruf unter Verwendung des Befehls „require“.

```
<?php
$conn = @mysql_connect("localhost","root");
mysql_select_db("cms4ds",$conn);
$editor='cms4ds_editor.php?action=';
?>
```

In der zweiten Zeile werden Benutzername und Passwort angegeben. **Dieser Teil muss also in der Regel angepasst werden!**

Die dritte Zeile enthält den Namen der Datenbank und die vierte Zeile den Link zum Editor, welcher auch angepasst werden kann, wenn benötigt.

Doch nun zur eigentlichen Abfrage, welche in cms4ds.php eingefügt wird:

```
<?php
function iso8859_mydecode($html){   // simple translation of german letters
  $a = array('ä'=>"xC3xA4",'ö'=>"xC3xB6",'ü'=>"xC3xBC",
             'Ä'=>"xC3x84",'Ö'=>"xC3x96",'Ü'=>"xC3x9C",'ß'=>"xC3x9F");
  return strtr($html,$a);
}
function utf8_mydecode($html){      // reverse simple translation of german letters
  $a = array("xC3xA4"=>'ä',"xC3xB6"=>'ö',"xC3xBC"=>'ü',
             "xC3x84"=>'Ä',"xC3x96"=>'Ö',"xC3x9C"=>'Ü',"xC3x9F"=>'ß');
  return strtr($html,$a);
}
$t0=sscanf(microtime(),"%f %s");
require "cms4ds.php.inc";
$where=
  preg_replace('/([^a-zA-Z%'0-9äöüÄÖÜß_(), ])/','',utf8_mydecode($_REQUEST['where']));
$recs = mysql_query("select * from cms4ds_content where find_in_set('"
  .$where."',replace(Rubrik,' ',',')) order by Sequence, Titel_Datum desc");
$out='';
while ($rec = mysql_fetch_assoc($recs)) {
$out.='<div class="content" style="display:block"><h3 onclick="bookmark(''
.$rec['Titel'].'')">'.$rec['Titel'].'</h3>'
.($rec['Hide_Date']=='y'?'':'<i>'.substr($rec['Titel_Datum'],0,10))
.($rec['Autor']==''?'':' - ').$rec['Autor'].'</i><p>'.$rec['Content'].'</p>';
if (substr($_SERVER['REMOTE_ADDR'],0,7)=='192.168') {
  $out.='<p style="position:absolute;top:150px;margin-left:-50px">'
  .'<a target="_blank" onclick="window.open(''.$editor
  .'new','','top=200,left=300,width=600,height=700')">new/lt;/a><br/>'
```

```
  .'<a target="_blank" onclick="window.open(''.$editor
  .'search&Content_ID='.$rec['Content_ID']
  .'','','top=120,left=250,width=616,height=540')"%gt;edit</a></p>';
  }
$out.='</div>';
}
@mysql_close($conn);
$t1=sscanf(microtime(),"%f %s");
print $out;
?>
<!--#set var="t" value="<?php print 'PHP: '.($t1[0]-$t0[0]).' sek. | ' ?>" -->
```

Einmal Schritt für Schritt:

Zunächst werden die „deutschen Sonderzeichen“ (ä,ö,ü,Ä,Ö,Ü) in Code umgewandelt, damit es nicht zu Problemen kommt.

Dann wird eine Variable mit Zeitstempel erstellt und am Ende mit einer zweiten abgeglichen um die Zeit des Zugriffs messen zu können.

Nun folgt eine weitere Zeichenüberprüfung. Doch dieses Mal werden Probleme nicht in Codes umgewandelt, sondern rigoros gelöscht. Also: Vorsicht mit Sonderzeichen.

Und endlich folgt dann der Datenbankzugriff.

Bei einer Abfrage wird die Sortierung zuerst versucht anhand des „sequenc“-Wertes versucht. Sollten alle diese Werte gleich sein, wird auf das Datum zugegriffen.

Am Ende sieht dann die Struktur eines einzelnen Beitrages aus wie folgt:

```
<div class="content" style="display:block" >
// damit auch ein Bookmark-Integration erfolgen kann, wird der Titel hierfür vorbereitet
<h3 onclick="bookmark('Titel')">Titel</h3>
//falls das Hide_Date ungleich y ist, dann wird zudem das Titel_Datum ausgeben
//falls der Autor existiert wird er auch mit ausgegeben, getrennt mit einem '-'
<i>Titel_datum - Autor</i>
<p>Content</p>
//falls die Ausgabe im lokalen Netz erfolgt (Adress-Raum 192.168.x.x),
//dann wird der Editor-Aufruf eingebaut
<p style="position:absolute;top:150px;margin-left:-50px">'
  .'<a target="_blank" onclick="window.open(''.$editor.'new','','top=200,
    left=300,width=600,height=700')">new</a><br/>'
  .'<a target="_blank" onclick="window.open(''.$editor.'search&Content_ID='
  .$rec['Content_ID'].'','','top=120,left=250,width=616,height=540')">edit</a></p>
</div>
```

Ich denke die Kommentare erklären alles von selbst.

Itari hat auch an eine Datenbank zum erfassen aller Besucher gedacht. Diesen Teil möchte ich mir hier aber sparen, da er nicht elementar für die Funktionsweise des CMS ist. Nur noch einmal der kurze Hinweis: Wer diese Funktion in vollem Umfang nutzen möchte, muss vorher die entsprechende Datenbank erstellen.

g. RSS

Jede moderne Website stellt einen bereit. RSS-Feeds. Sie werden verwendet um sich regelmäßig ändernde Seiten, welche Beiträge nach Zeit geordnet enthalten, mit externen Programmen ansehen zu können und diese zu abonnieren. Wie wir bereits gesehen haben, ist itaris CMS moderner als so manch aufgeblasenes fertig-CMS. Daher darf natürlich auch ein RSS-Feed nicht fehlen.

Den entsprechenden Link in der header.html haben wir ja bereits gesehen. Dieser führt uns zur Datei cms4ds_rss.php.

```
<?php
function iso8859_mydecode($html){         // simple translation of german letters
  $a = array('ä'=>"xC3xA4",'ö'=>"xC3xB6",'ü'=>"xC3xBC",
             'Ä'=>"xC3x84",'Ö'=>"xC3x96",'Ü'=>"xC3x9C",'ß'=>"xC3x9F");
  return strtr($html,$a);
}
header('Content-Type: application/xml');
print '<?xml version="1.0" encoding="UTF-8"?>'; ?>
<?php
print '<rss version="0.91"><channel><title>cms4ds-News-Feed</title>'
      .'<link>http://'.$_SERVER['HTTP_HOST'].$_SERVER['PHP_SELF'].'</link>'
      .'<description></description><language>de</language>';
require "cms4ds.php.inc";
$recs = mysql_query("select Titel,Rubrik,Titel_Datum from cms4ds_content
                                order by Titel_Datum desc limit 2");
while ($rec = mysql_fetch_assoc($recs))
      print '<item><title>'.iso8859_mydecode($rec['Titel']).'</title><link>'
            .'http://'.$_SERVER['HTTP_HOST'].dirname($_SERVER['PHP_SELF']).'/'
            .$rec['Rubrik'].'.html?id='
            .rawurlencode(iso8859_mydecode($rec['Titel'])).'</link></item>';
print '</channel></rss>';
?>
```

Dieser php-Aufruf gibt eine XML-Datei nach RSS-Standard 0.91 wieder. Der Aufruf ähnelt dem eines normalen Such-Aufrufs bei dem alle Beiträge nach Datum geordnet werden. Im Quelltext sind zwei Teile rot hervorgehoben. Diese sollte man an die eigenen Bedürfnisse anpassen. Der erste stellt dabei den Namen des Feeds und die zweite die Anzahl der darzustellenden Beiträge dar.

```
<?xml version="1.0" encoding="UTF-8"?>
<rss  version "0.91">
 <channel>
  <title>cms4ds-News-Feed</title>
  <link>http://syno/cms4ds/cms4ds_rss.php</link>
  <description></description>
  <language>de</language>
  <item>
   <title>cms4ds als RSS-Feed</title>
   <link>http://syno/cms4ds/rss.html?id=cms4ds%20als%20RSS-Feed</link>
  </item>
  ...
 </channel>
</rss>
```

Der obige Text stellt dar, wie eine fertige XML-Seite aussehen könnte.

Wer RSS-Feeds von anderen Seiten einbinden möchte, kann auf bereits vorhandene Funktionen zurückgreifen, welche itari uns mitliefert. Mehr Informationen dazu, gibt es auf seiner Seite.

Zur Suche auf der eigenen Seite unter Verwendung des OpenSearchDescription-Protokolls gibt es auch eine eigene XML-Datei. Diese wird unter anderem von Browsern genutzt, um die Suchanfragen aus den kleinen Fenstern rechts oben im Browser durchzuführen. Auch auf diese möchte ich hier allerdings nicht näher eingehen.

h. Tools

Wer sich bisher aufmerksam durch die Quelltexte gearbeitet hat, dem werden ein paar merkwürdige Zeilen aufgefallen sein, auf die ich nicht genau eingegangen bin.

```
//falls die Ausgabe im lokalen Netz erfolgt (Adress-Raum 192.168.x.x),
//dann wird der Editor-Aufruf eingebaut
<p style="position:absolute;top:150px;margin-left:-50px">'
  .'<a target="_blank" onclick="window.open(''.$editor.'new','','top=200,
    left=300,width=600,height=700')">new</a><br/>'
  .'<a target="_blank" onclick="window.open(''.$editor.'search&Content_ID='
  .$rec['Content_ID'].'','','top=120,left=250,width=616,height=540')">edit</a></p>
</div>
```

Die Datei cms4ds_editor.php enthält einen Texteditor, mithilfe dessen sich die einzelnen Seiten unkompliziert bearbeiten lassen. Wenn eine Überprüfung der IP ergibt, dass die Anfrage von einem Computer kommt, dessen Adresse mit 192.168 beginnt, wird als weiteres Fenster der Editor geöffnet. Alle LAN-Adressen beginnen so, sollten sie nicht stark an ihren Netzwerkeinstellungen gedreht haben.

cms4ds - Mozilla Firefox
00023 Web-Editor für das cms4ds
Web-Editor für das cms4ds
2008-11-06 00:00:00
hide (y):
sequence-#: 0
Für die Erfassung des dynamischen Contents steht ein kleiner Web-Editor namens cms4ds_editor.php zur Verfügung. In der Datei cms4ds.php wird bei der Ausgabe der Beiträge geprüft, ob der Zugriff aus dem lokalen Netzwerk stammt (IP-Adress-Raum 192.168.x.x) und ein Link für den Aufruf des Editors so eingebaut, dass man direkt im Editor den auf dem Beitrag steht. Das Editor-Fenster enthält ein Auswahl-Feld, mit dem man auf alle Beiträge zugreifen kann. Mit einem Doppelklick auf das Content-Feld wird ein Voransicht eingeblendet, die auf dem aktuelle CSS basiert. Die Zuordnung zu einer Rubrik erfolgt in der untersten Zeile.
tools
cms4ds database files javascript rss search stylesheet tools
remove
save
Fertig
Notizbuch öffnen

Wie bereits beschrieben, wird eine zweite Datenbank angelegt, wo alle Zugriffe gespeichert werden. Um aber nicht diese Datenbank kompliziert auslesen zu müssen, gibt es auch hierfür das passende PHP-Skript. Der Aufruf erfolgt über cms4ds_showlog.php

8. Andere Anwendungen auf der DS („3rd-Party-Apps“)

Zu diesem Kapitel existiert auch ein „Integration-Guide“ von Synology, welchen man sich zusätzlich durchlesen sollte, bevor man etwas ändert.

a. Vor dem Modden!

Bevor ich mit irgendeiner Anleitung beginne, welche sich mit der Veränderung von Soft- oder Hardware befasst (ich werde mich allerdings auf Software beschränken), muss ich noch auf die „Nebenwirkungen“ hinweisen. **Weil wir jetzt Anwendungen von dritten verwenden, übernimmt Synology keine Garantie für eventuelle Probleme. Daher ist alles auf absolut eigene Gefahr!** Solange wir bei der Software bleiben, sollte nach einem erneuern der Firmware alles wieder beim alten sein. Doch dieser Vorgang ist immer mit dem Verlust aller bisher gespeicherten Daten verbunden. Daher ist ein Backup vor dem Modden ein Muss.

Die DiskStation kann aber nicht von Anfang an richtig mit unseren Modifikationen umgehen. Daher hat itari das SPK-Paket „init_3rdparty“ geschrieben. Einmal dieses ausgeführt, wird das Linux entsprechend vorbereitet. Nun können wir weitere Anwendungen installieren.

b. Toolchain

Von hier an werde ich mich mit Programmierung beschäftigen. Wer daher noch nie mit C gearbeitet hat, sollte hier aufhören und stattdessen ein dahingehendes Buch konsultieren, bevor man zurückkehrt.

Wie für Linux üblich, wird zur Programmierung eine Toolchain, also eine Sammlung von Programmierwerkzeugen, welche zum Erstellen von Programmen benötigt werden (Compiler, Debugger, Linker, Assembler, …) verwendet. Synology hat zu diesem Zweck eine eigene Toolchain erstellt um die Entwicklung von Anwendungen zu erleichtern. Diese kann im Internet heruntergeladen werden: http://download.synology.com/toolchain. Diese sollte man direkt auf ein Linux-System laden und dort entpacken. Nun kann das Programmieren beginnen.

c. Bekannte 3rd-Party-Anwendungen

Da diese Liste hier zu unübersichtlich geworden wäre, wurde sie in das Wiki verschoben:
http://synology-wiki.de/

9. Nützliche Links

Die folgenden Links enthalten nützliche Informationen zum Thema Synology DiskStations, Linux-Entwicklung und allen anderen Dingen die hier besprochen wurden.

- Deutsches Synology-Forum: http://synology-forum.de
- Internationales Forum: http://synology.com/forum
- Deutsches Synology-Wiki: http://synology-wiki.de
- Internationales Wiki: http://synology.com/wiki
- Synology Support inklusive Firmware-Updates, Kompatibilitätslisten und FAQ: http://synology.com/enu/support/index.php
- itaris kleine Seite zum Thema DiskStation und insbesondere deren Modifikationen: http://itari.synology-ds.de/ und http://itari.syno-ds-de/